



Welcome to the Presentation



Workshop on Genomics and Bioinformatics

Gene-Expression Data Analysis Lecture-1

August 3-5, 2018

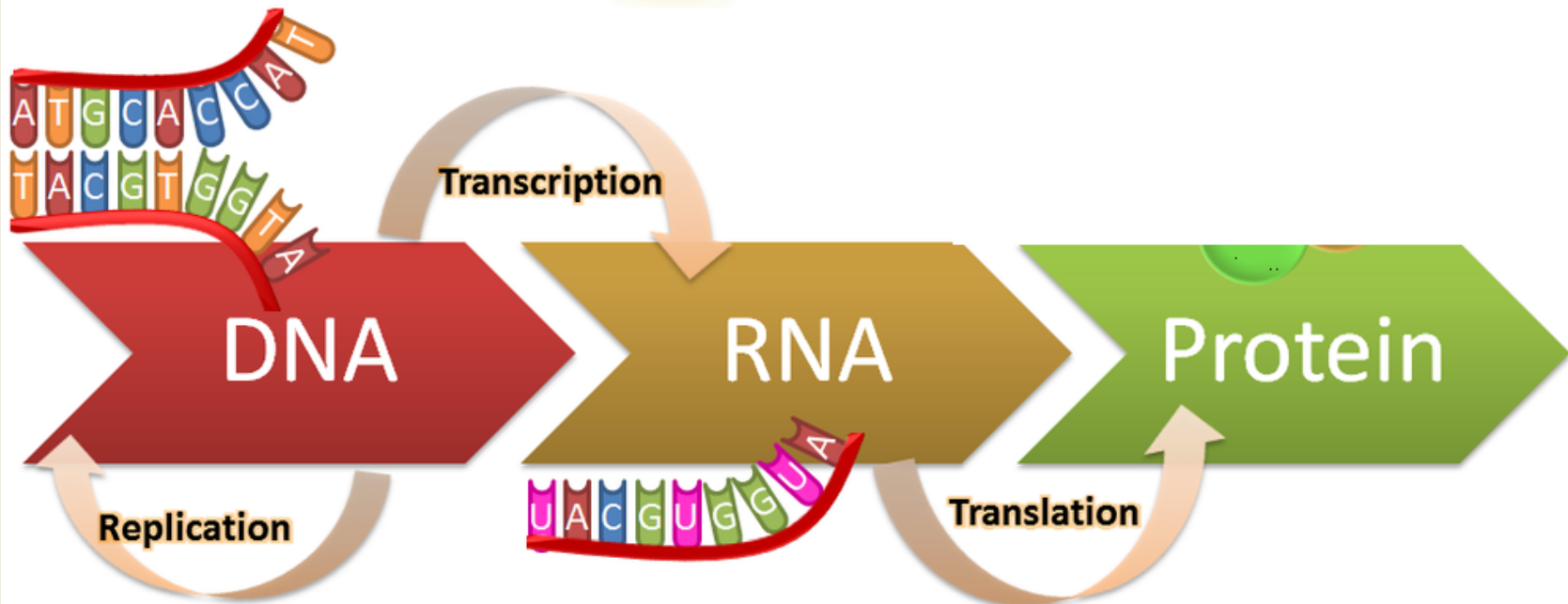
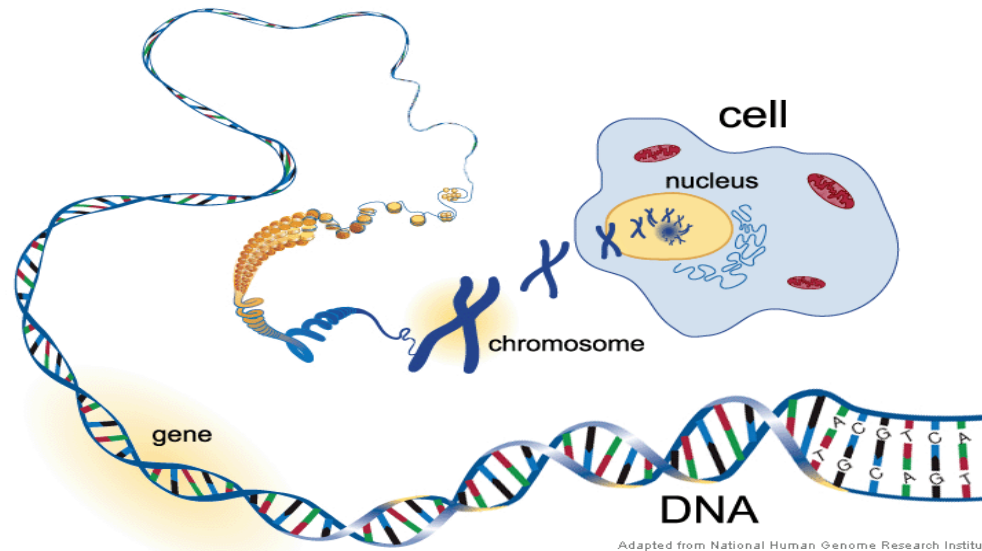
Nishith Kumar

Assistant Professor, Department of Statistics
Bangabandhu Sheikh Mujibur Rahman Science and Technology University.
Gopalganj, Bangladesh

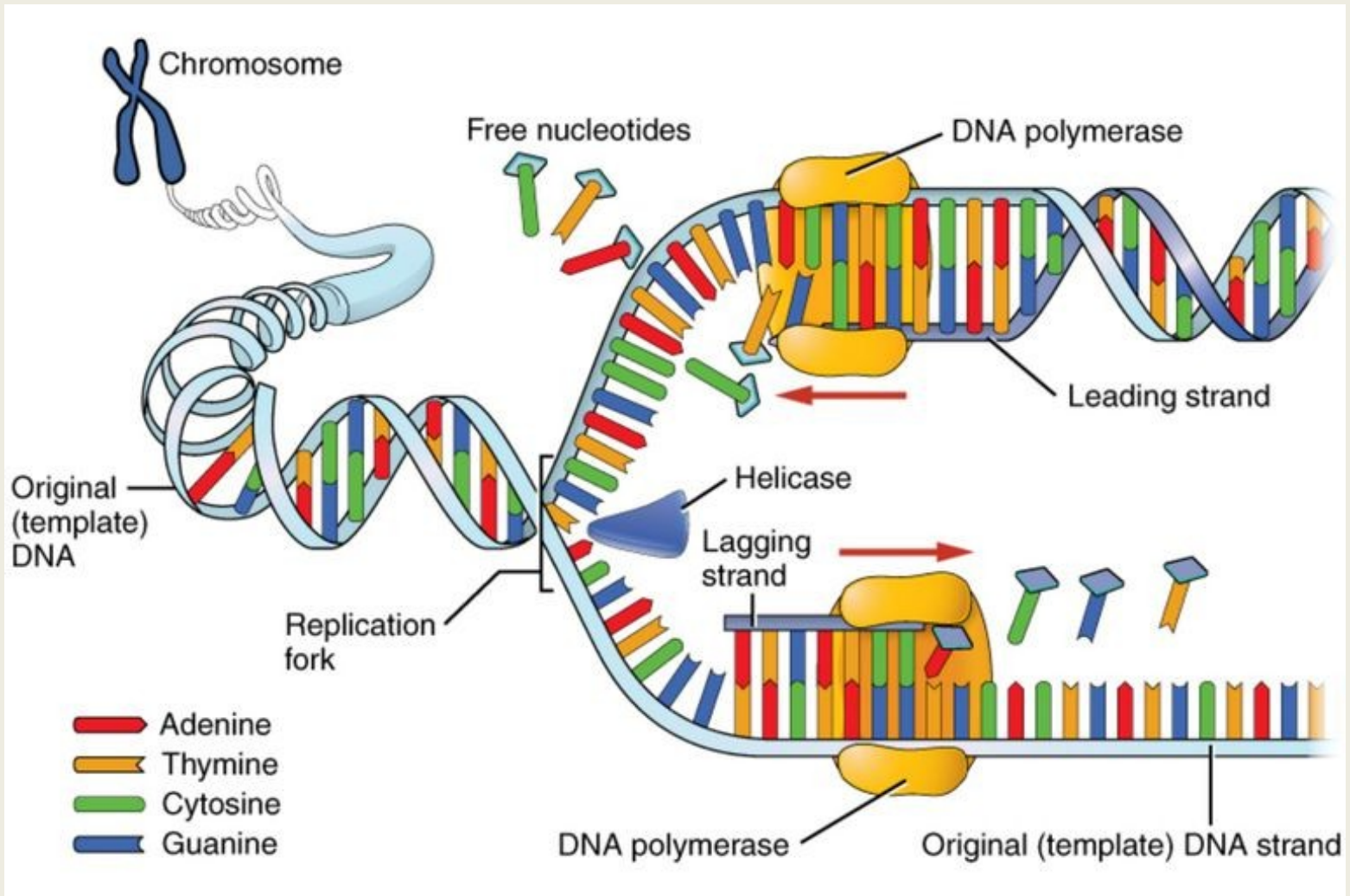
Overview

- Central Dogma of Molecular Biology
- Gene Expression
- Transcriptomics
- Application of Microarray gene expression analysis
- Steps to Generate the Gene Expression Data Using cDNA Microarray Technology
- Data Quantification
- Pipeline of microarray gene expression data analysis
- Gene expression Dataset description
- Problems of microarray data
- Data Preprocessing

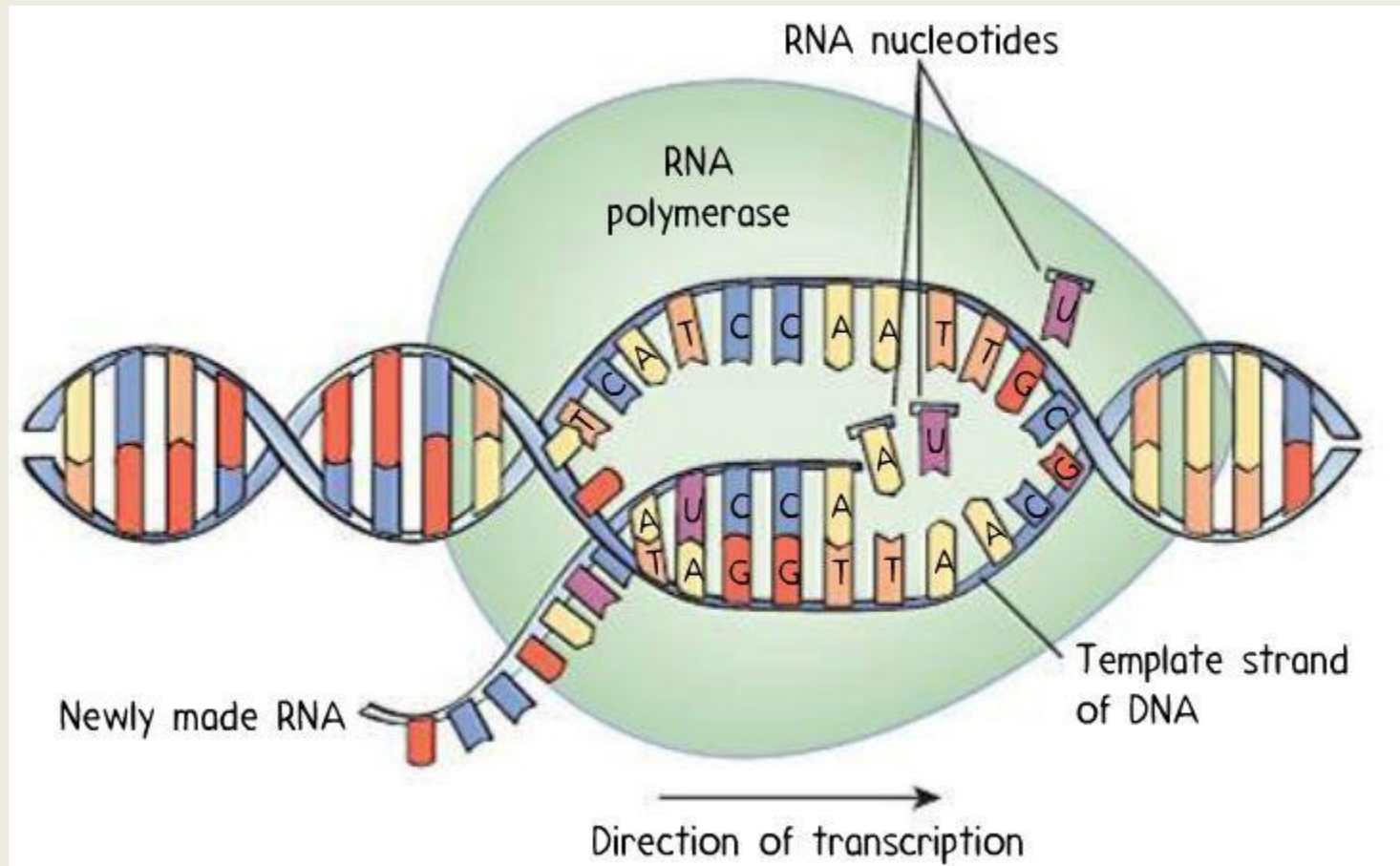
Central Dogma of Molecular Biology



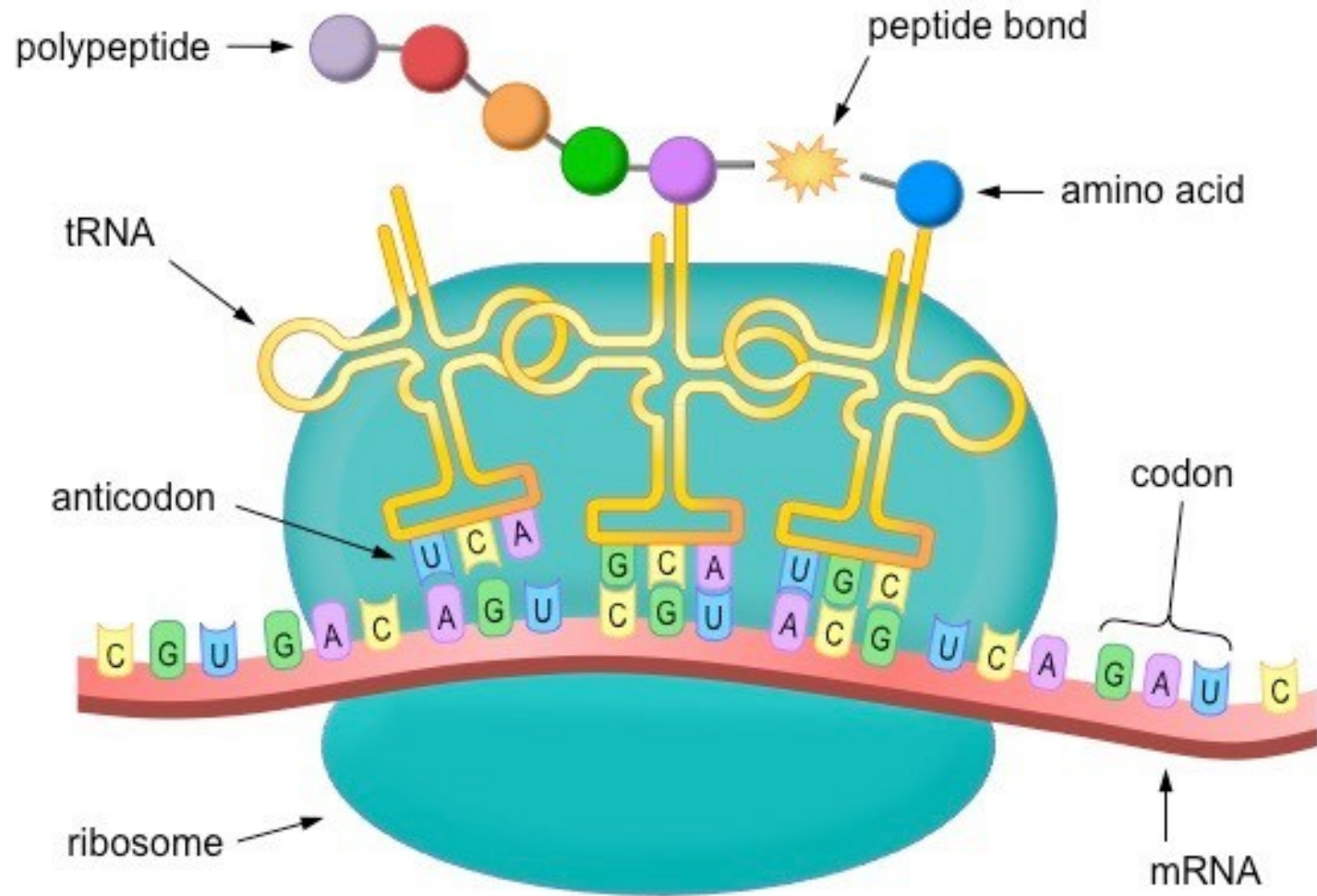
DNA Replication



Transcription

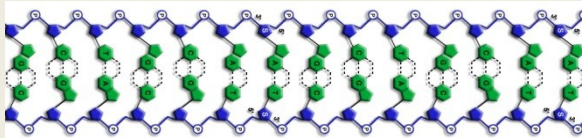


Translation



Central Dogma of Molecular Biology

DNA



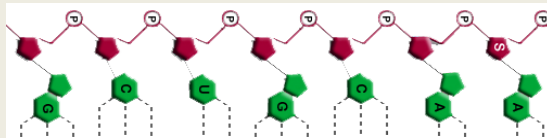
Genomics – 40,000 Genes



Transcription



RNA



Transcriptomics – 1,00000 Transcripts



Translation

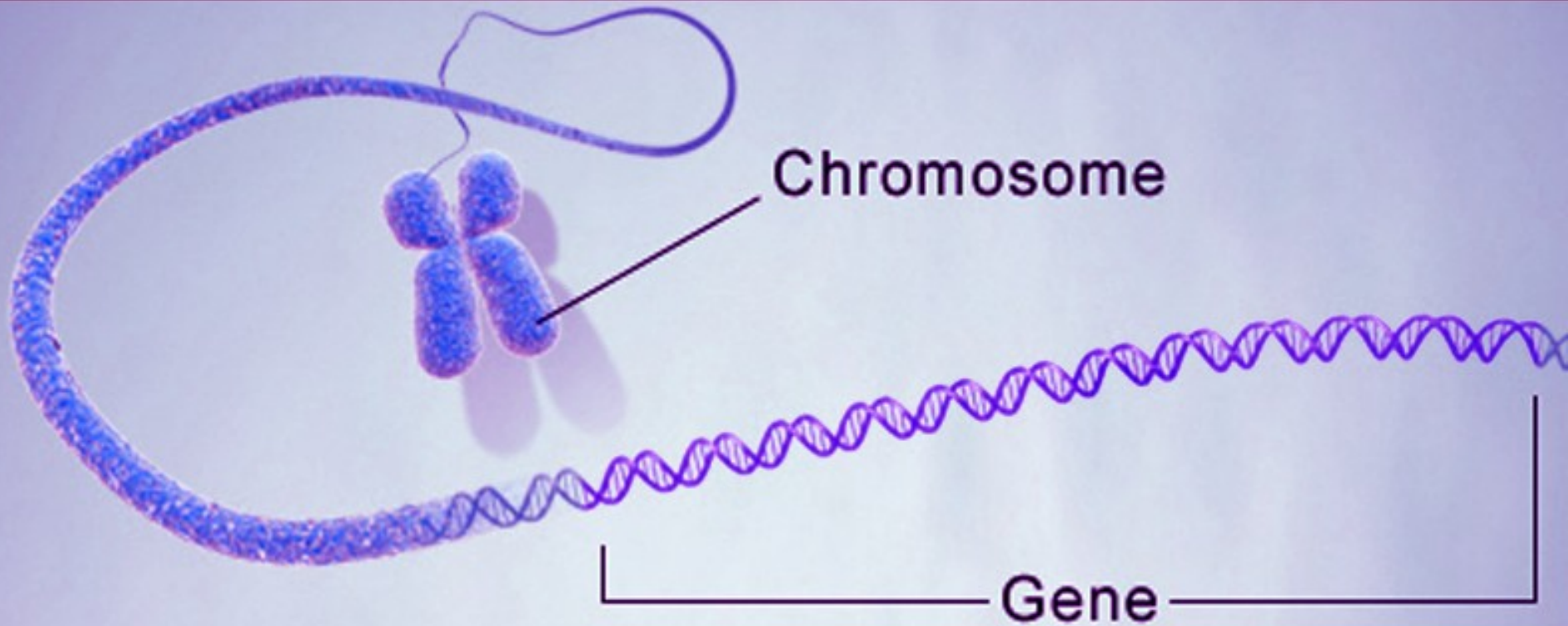


Protein



Proteomics – 10,00,000 Proteins

GENE



Gene is a basic unit of heredity which is transferred from a parent to offspring and is held to determine some characteristic of the offspring.

Gene Expression

Gene expression is the process by which the instructions in our DNA are converted into a functional product, such as a protein.

Gene expression is a tightly regulated process that allows a cell to respond to its changing environment.

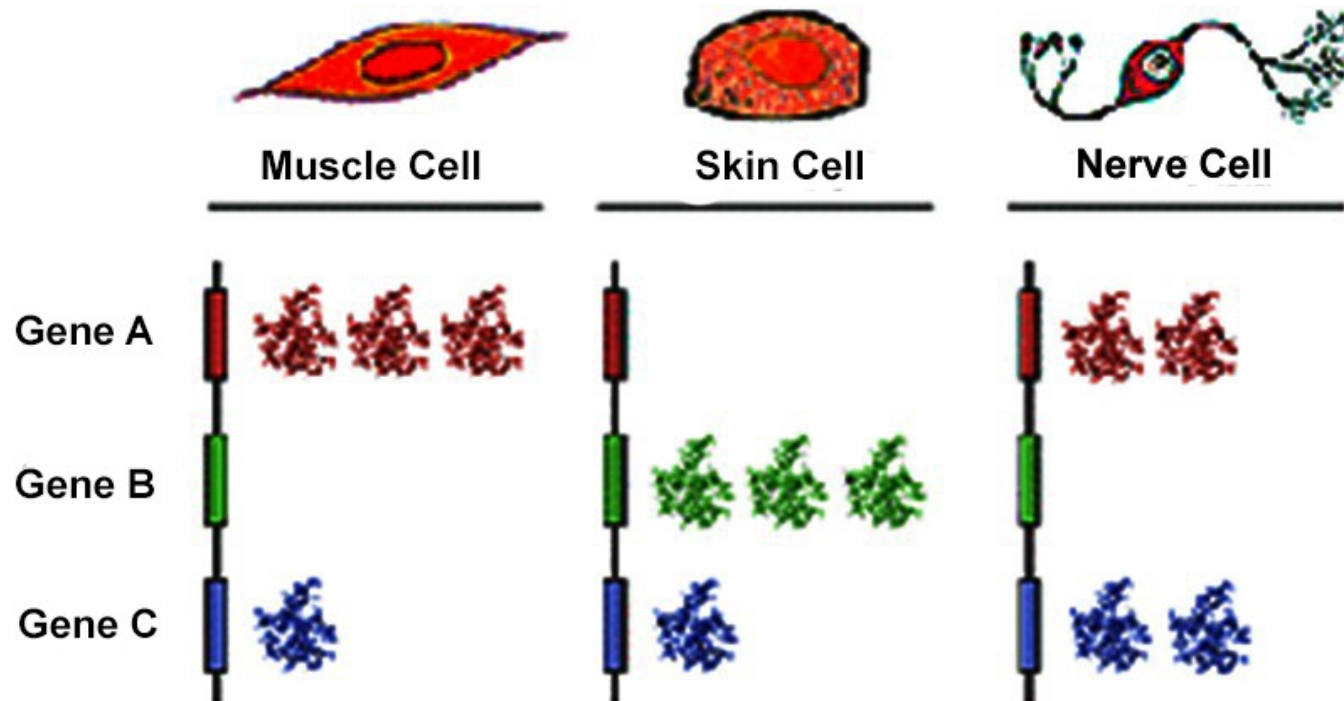
It acts as both an on/off switch to control when proteins are made and also a volume control that increases or decreases the amount of proteins made.

Gene Expression

1. A human organism has over 250 different cell types (e.g., muscle, skin, bone, neuron), most of which have identical genomes, yet they look different and do different jobs
2. It is believed that less than 20% of the genes are ‘expressed’ (i.e., making RNA) in a typical cell type
3. Apparently the differences in gene expression is what makes the cells different

Gene Expression in Different Position

Cells and Gene Expression



Some questions for the golden age of genomics

- How gene expression differs in different cell types?
- How gene expression differs in a normal and diseased (e.g., cancerous) cell?
- How gene expression changes when a cell is treated by a drug?
- How gene expression changes when the organism develops and cells are differentiating?
- How gene expression is regulated – which genes regulate which and how?

Transcriptomics

Definition of Transcriptomics

Transcriptomics, the study of RNA in any of its forms. The transcriptome is the set of all RNA molecules, including mRNA, rRNA, tRNA, and other non-coding RNA produced in one or a population of cells.

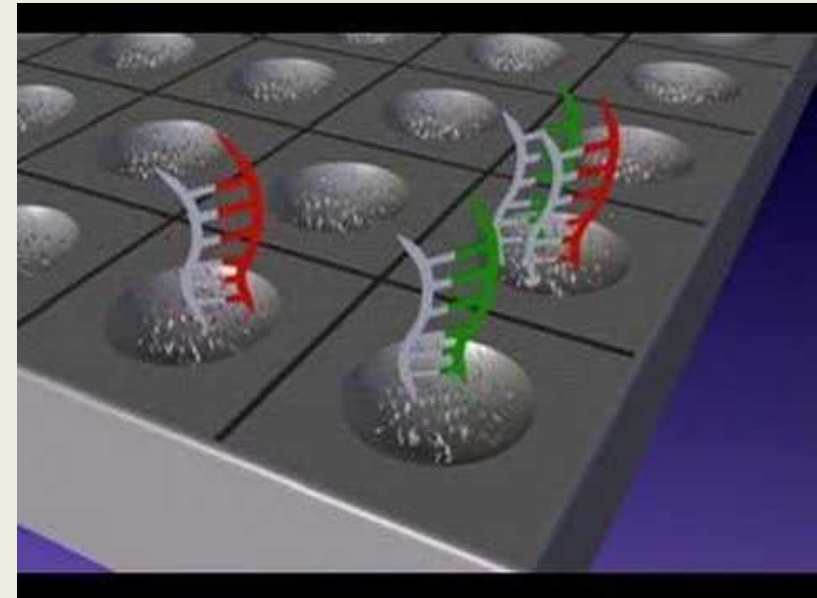
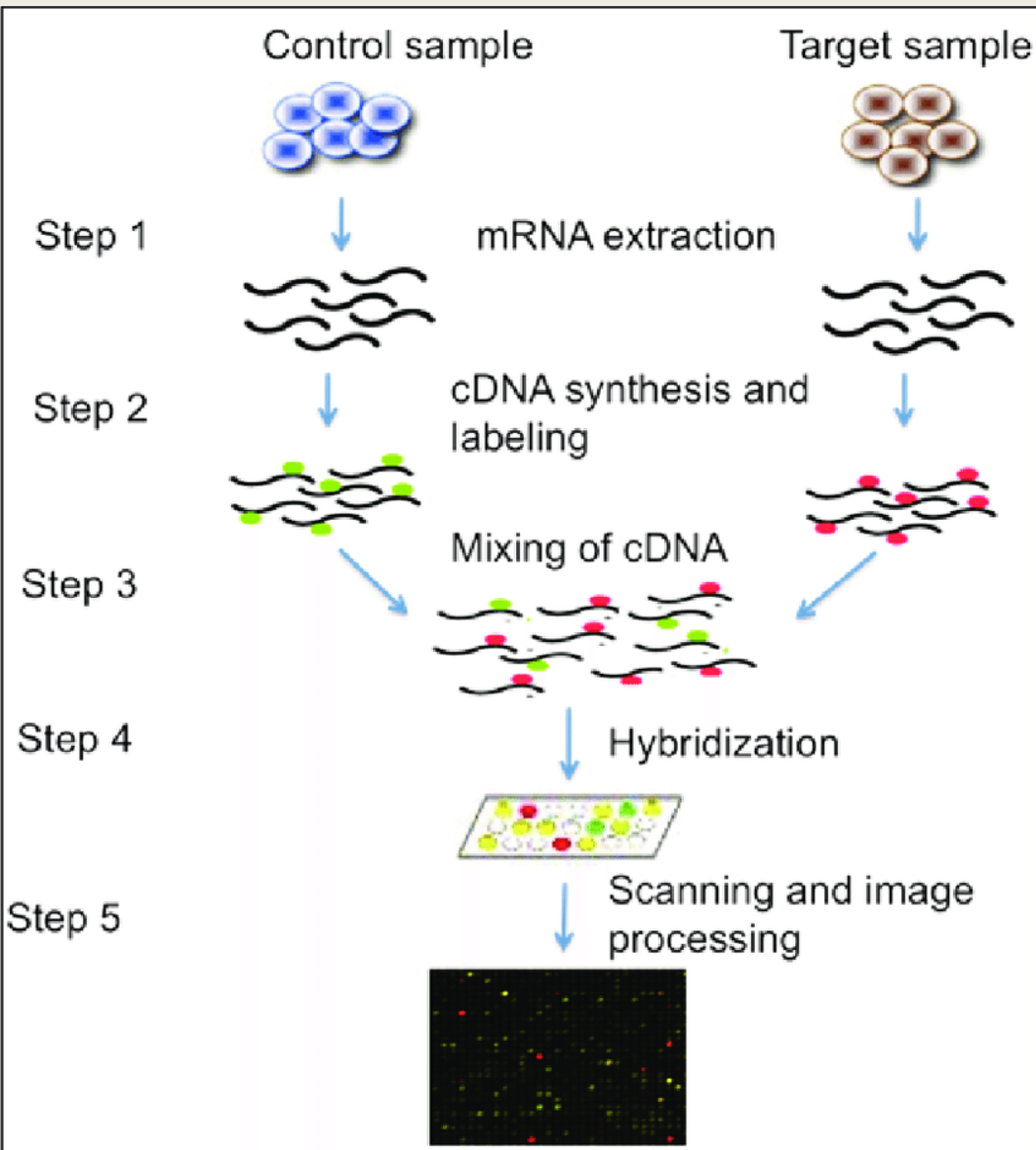
Objectives of Transcriptomics

- To identify differentially expressed genes.
- To show changes in gene expression in a diseased cell.
- To determine the transcriptional structure of genes.
- To provides data for thousand of genes.
- To detect Differentially Expressed (DE) genes.

Application of Microarray Gene Expression Data Analysis

1. Vaccine Candidate Identification
2. Gene Expression Profiling
3. Polymorphism/SNP detection
4. Host Pathogen Identification
5. Drug Target Identification
6. Disease Diagnosis
7. Genetic Engineering
8. Toxicogenomics Research
9. Differentially Expressed Gene Identification
10. Response Measure of Environmental Factors

Steps to Generate the Gene Expression Data Using cDNA Microarray Technology



Steps to Generate the Gene Expression Data Using cDNA Microarray Technology

Step-1:

Since we are interested in comparing gene expression, one sample usually serves as control, and another sample would be the experiment (healthy vs. disease, etc). We need to isolate/extract mRNA in this step.

Step-2:

In order to detect the transcripts by hybridization, they need to be labeled, and because starting material maybe limited, an amplification step is also used. Labeling usually involves performing a reverse transcription (RT) reaction to produce a complementary DNA strand (cDNA) and incorporating a florescent dye that has been linked to a DNA nucleotide, producing a fluorescent cDNA strand. Disease and healthy samples can be labeled with different dyes and co-hybridized onto the same microarray in the following step.

Steps to Generate the Gene Expression Data Using cDNA Microarray Technology

Step-3: This step mixing the disease and healthy samples.

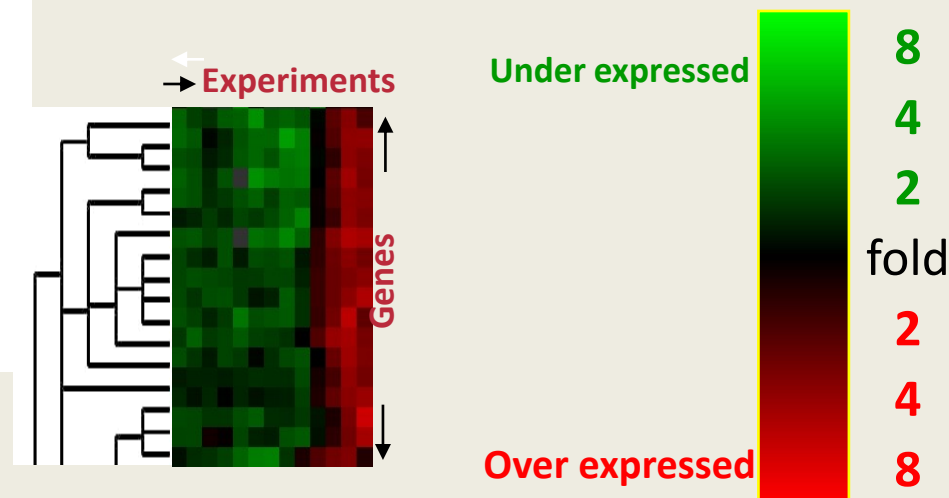
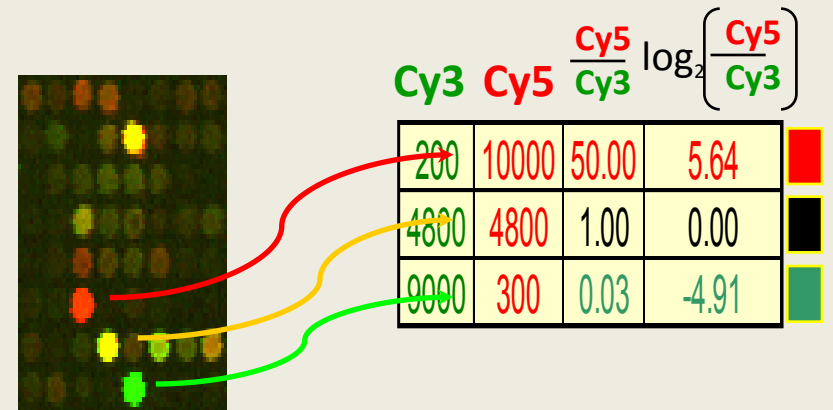
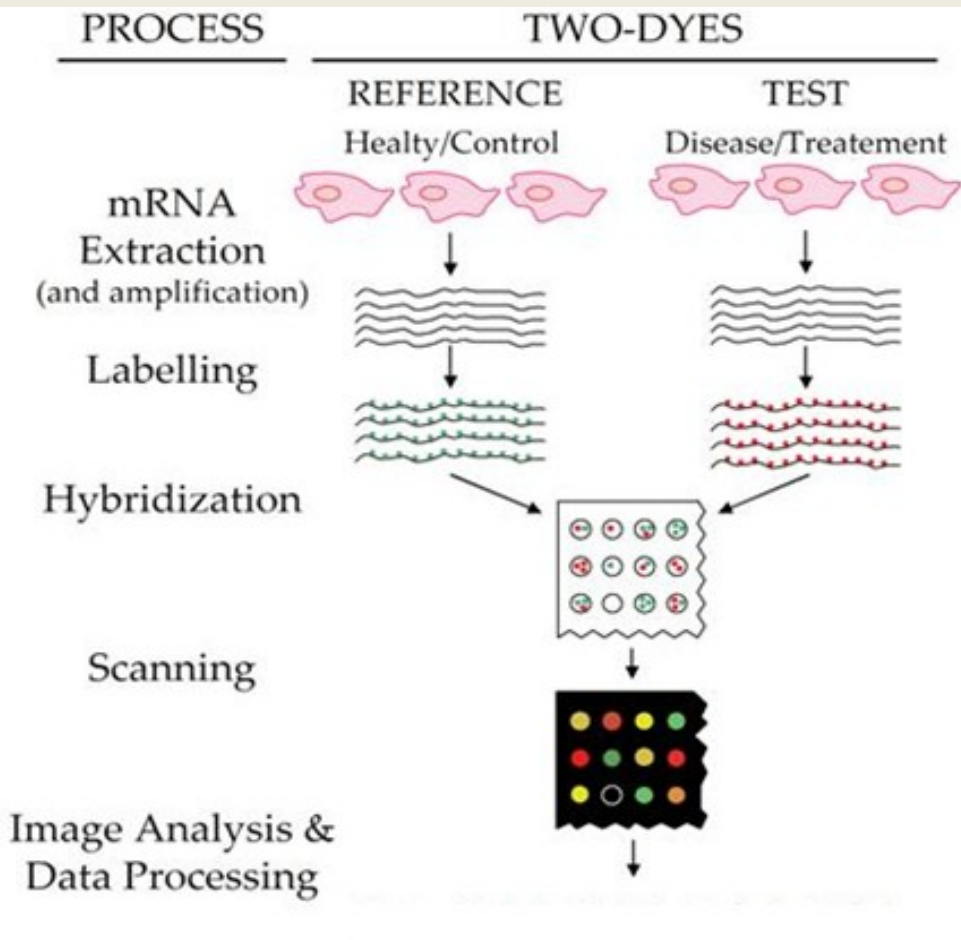
Step-4:

This step involves placing labeled cDNAs onto a DNA microarray where it will hybridize to their synthetic complementary DNA probes attached on the microarray. A series of washes are used to remove non-bound sequences.

Step-5:

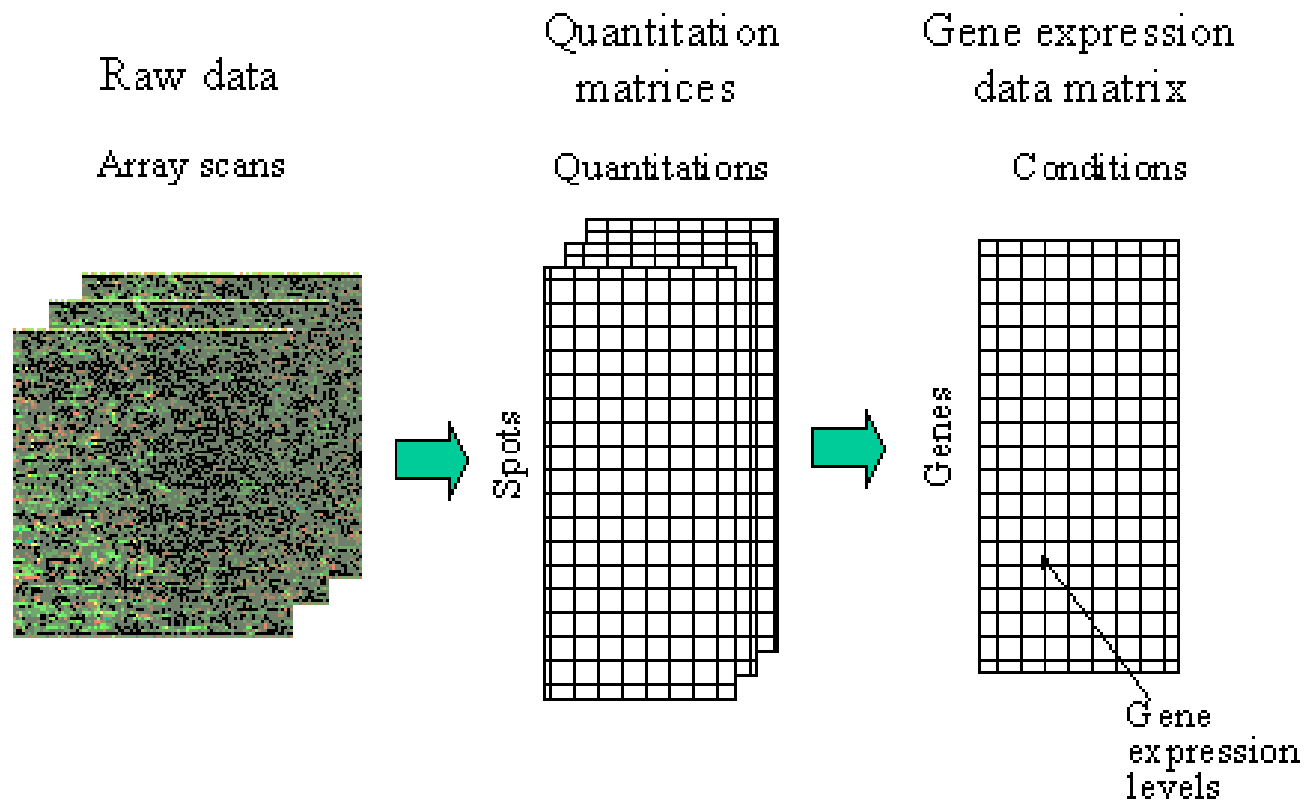
The fluorescent tags on bound cDNA are excited by a laser and the fluorescently labeled target sequences that bind to a probe generate a signal. The total strength of the signal depends upon the amount of target sample binding to the probes present on that spot. Thus, the amount of target sequence bound to each probe correlates to the expression level of various genes expressed in the sample. The signals are detected, quantified, and used to create a digital image of the array.

Quantification

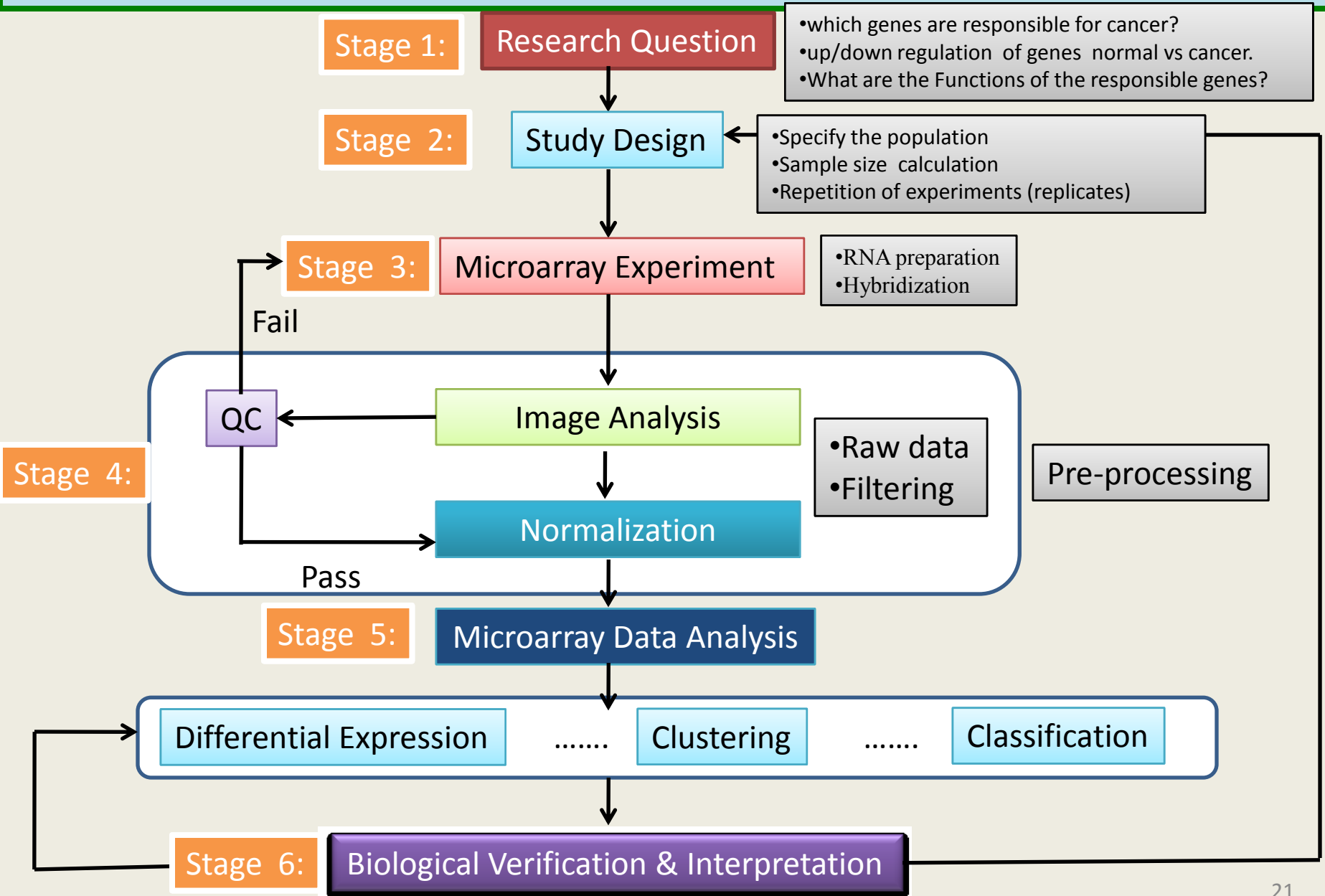


1	0.0	2.3	0.01	-1.51
2	5.1	-2.49	0.00	2.06
3	0.0	2.11	-3.04	0.52
4	-0.5	-1.03	-0.8	-0.6
5	-0.1	1.06	1.35	1.09

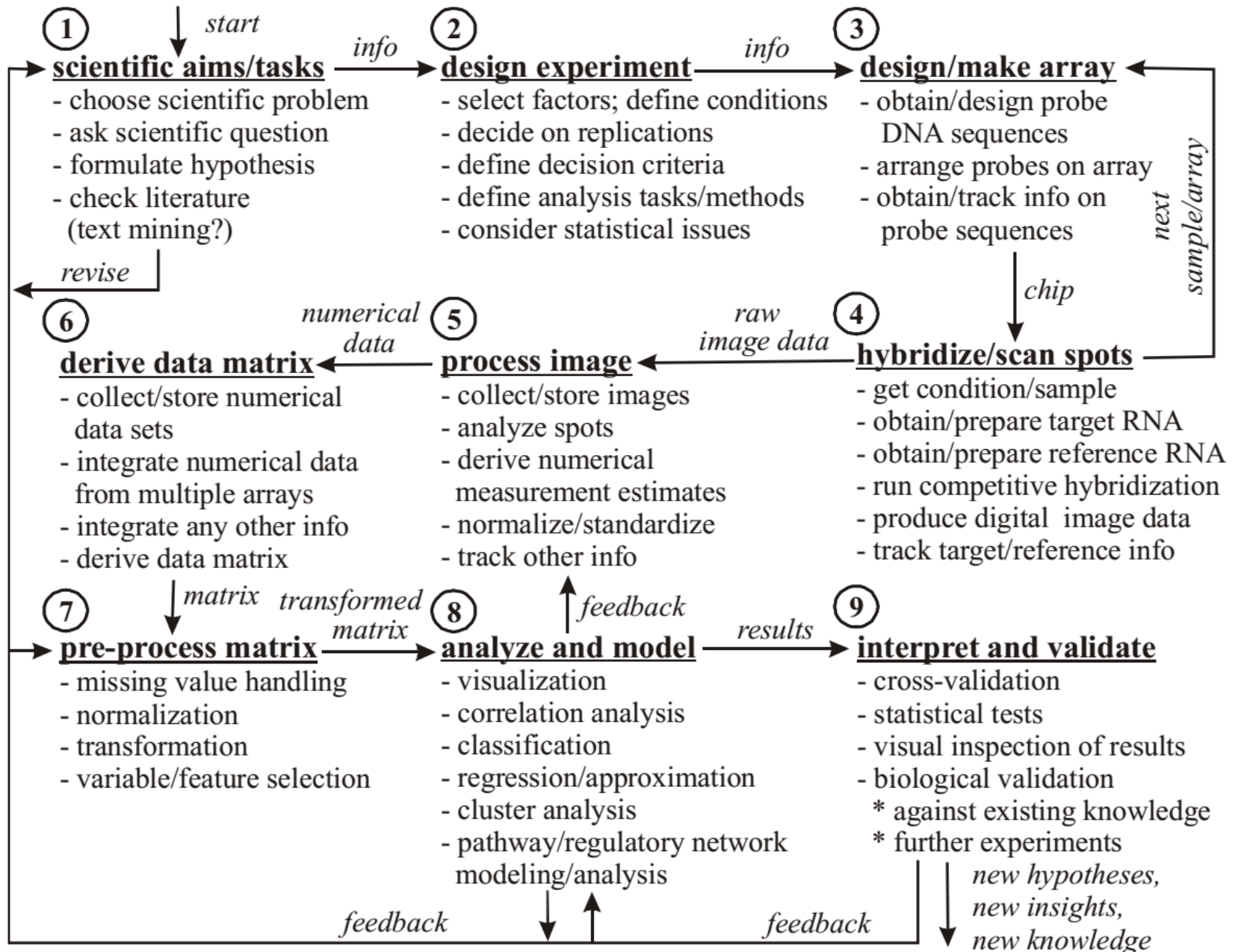
Microarray Data Processing and Analysis



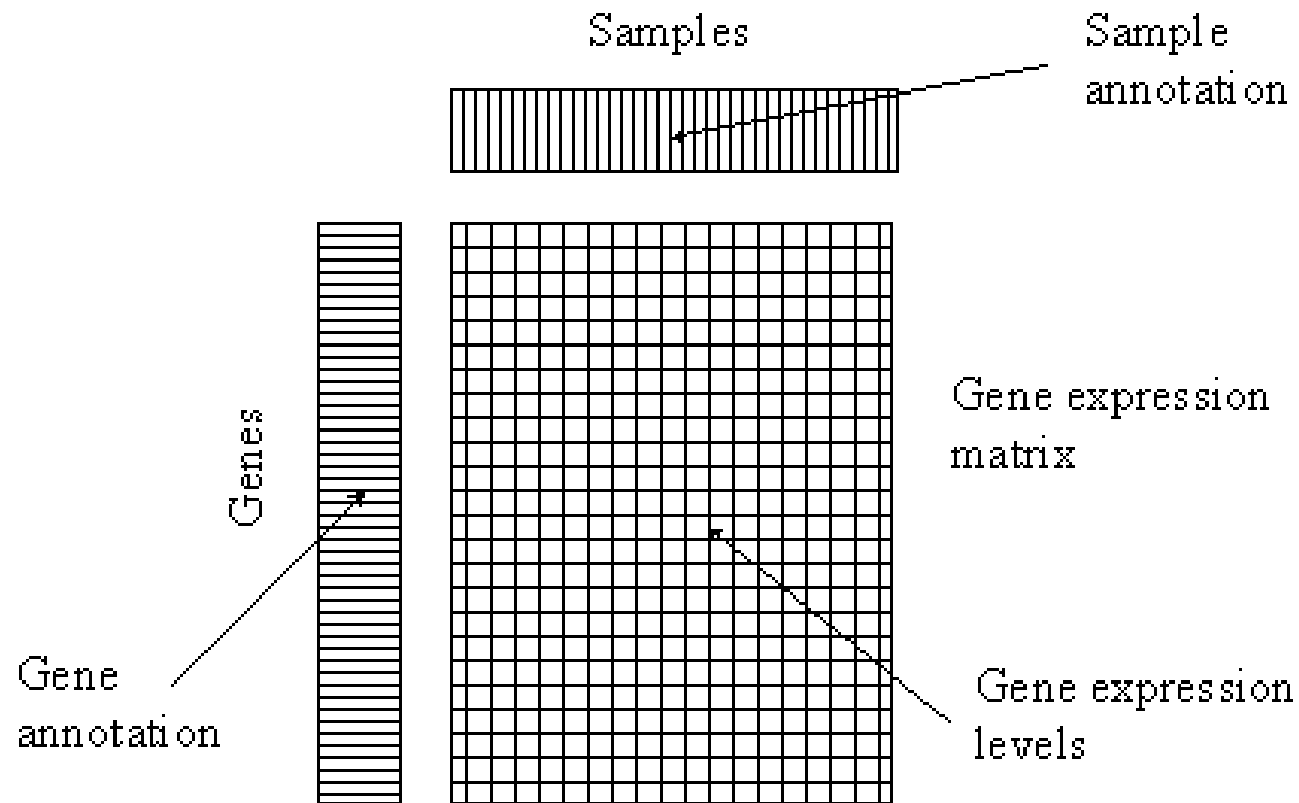
Pipeline of Microarray Gene Expression Data



Microarray Data Analysis Process



Gene Expression Data Description



Gene Expression Data Description

ID_REF	GSM258551	GSM258552	GSM258553	GSM258554	GSM258555
1007_s_at	9.129904553	9.843348744	97.306612	9.032164554	10.28179284
1053_at	8.034021594	7.973332021	8.834044796	7.723965395	9.040800258
117_at	3.56451954	4.994852417	5.066018134	4.958579594	4.951834721
121_at	4.746490311	5.197305596	5.234618433	6.078179973	52.056318
1255_g_at	2.320697769	2.3206977	2.25950441	2.262787006	2.207531074
1294_at	5.51915291	5.081257774	4.657257379	4.535683216	3.731919228
1316_at	3.339181611	2.934516219	3.007192191	3.167111656	2.711789535
1320_at	2.775394724	2.617096885	2.634558729	3.127495448	3.127495
1405_i_at	9.145519	9.145519209	9.159054057	7.770354395	6.380928488
1431_at	2.981326873	4.619667537	2.746463927	6.212399046	4.528498654

Problems of Microarray Data

ID_REF	GSM258551	GSM258552	GSM258553	GSM258554	GSM258555	GSM258556
1007_s_at	9.129904553	9.843348744	97.306612	9.032164554	10.28179284	9.154550982
1053_at	8.034021594	7.973332021	8.834044796	7.723965395	9.040800258	8.792375433
117_at	3.56451954	4.994852417	5.066018134	4.958579594	4.951834721	6.428273651
121_at	4.746490311	5.197305596	5.234618433	6.078179973	52.056318	5.009848299
1255_g_at	2.320697769	NA	2.25950441	2.262787006	2.207531074	2.322618304
1294_at	5.51915291	5.081257774	4.657257379	4.535683216	3.731919228	4.964671509
1316_at	3.339181611	2.934516219	3.007192191	3.167111656	2.711789535	2.958602178
1320_at	2.775394724	2.617096885	2.634558729	3.127495448	NA	2.598406037
1405_i_at	NA	9.145519209	9.159054057	7.770354395	6.380928488	8.149709807
1431_at	2.981326873	4.619667537	2.746463927	6.212399046	4.528498654	4.008139992

Why do we need data preprocessing for microarray gene expression data?

1. Huge systematic variation
2. Dataset may contains 10% to 40 % missing values.
3. Dataset also contain outliers/noise.

Therefore, Data Pre-processing is very much important to solve the above problems

Data Preprocessing Techniques

1. Shifting, Scaling, Standardization, Normalization and Transformation techniques
2. Outlier / Noise handling techniques
3. Missing value imputation techniques

Shifting, Scaling, Standardization, Normalization and Transformation

Shifting:

Data shifting is a technique that transforms the data by adding or subtracting a constant to each member of a data set. If $\mathbf{x}$ is a data vector and k is a constant then one possible shifting of $\mathbf{x}$ is

$$\mathbf{x}_{new} = \mathbf{x} - k.$$

Scaling:

Data scaling is a technique that transforms the data by multiplying or dividing a constant to each member of a data set. If $\mathbf{x}$ is a data vector and k is a constant then one possible scaling of $\mathbf{x}$ is

$$\mathbf{x}_{new} = \mathbf{x} / k.$$

Shifting, Scaling, Standardization, Normalization and Transformation

Standardization:

Standardization transforms the dataset into zero mean and unit variance. If $\mathbf{x}$ is a data vector with mean $\bar{\mathbf{x}}$ and standard deviation s then one possible standardization of $\mathbf{x}$ is $\mathbf{x}_{new} = (\mathbf{x} - \bar{\mathbf{x}}) / s$.

Normalization:

Normalization is a technique that transforms the data into a range $[0, 1]$. One possible formula of normalization is

$$\mathbf{x}_{new} = (\mathbf{x} - x_{\min}) / (x_{\max} - x_{\min}).$$

Shifting, Scaling, Standardization, Normalization and Transformation

Transformation:

Data transformation is a mathematical operation on each observation. There are several transformations in the literature, e.g.

1. log transformation
2. sqareroot transformation
3. principal component analysis
4. furrier transformation etc.

Shifting, Scaling, Standardization, Normalization and Transformation

Methods	Formula
Centering	$\tilde{x}_{ij} = x_{ij} - \bar{x}_i$
Auto scaling	$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{s_i}$
Range scaling	$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{x_{i_{\max}} - x_{i_{\min}}}$

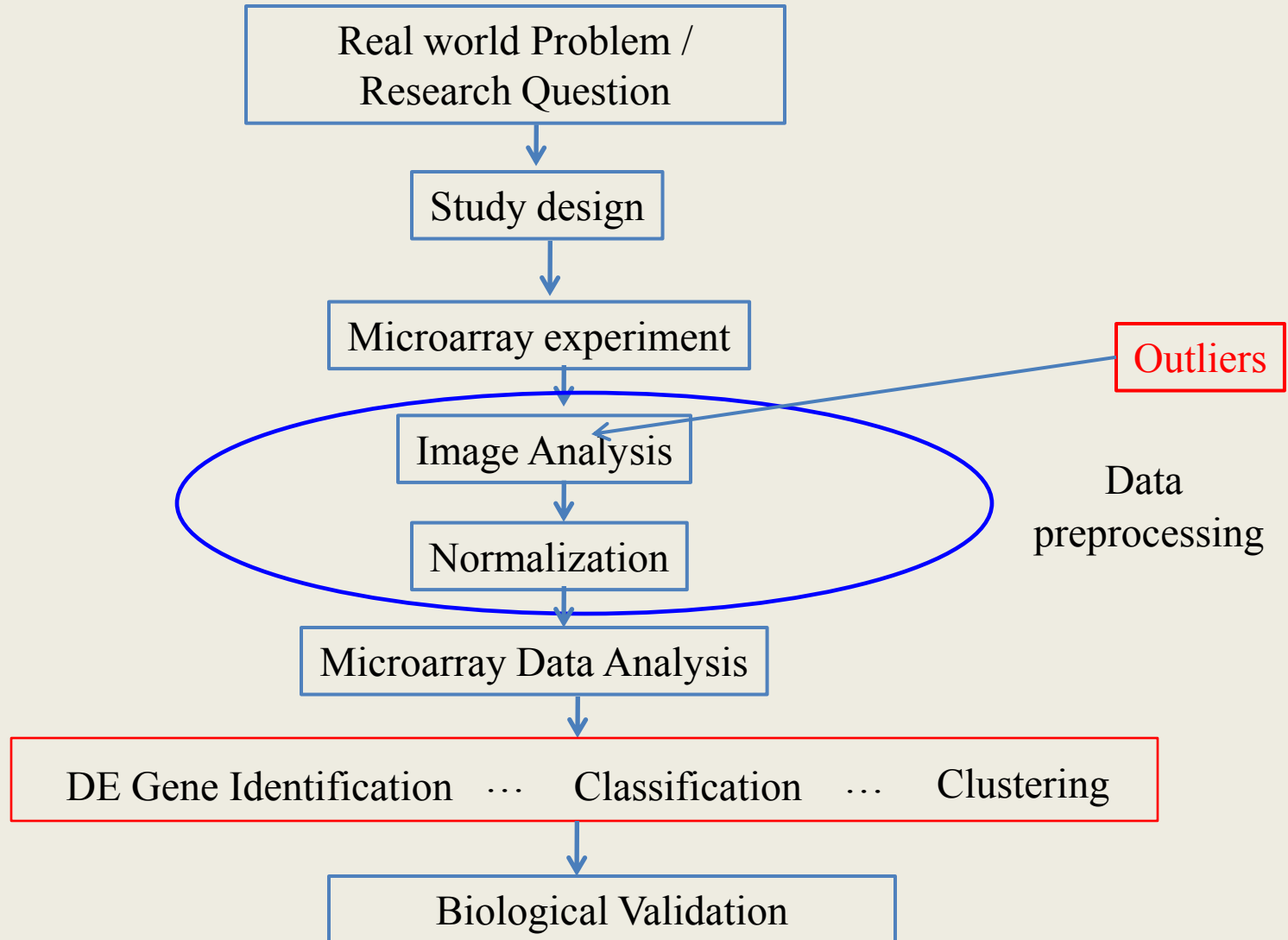
Shifting, Scaling, Standardization, Normalization and Transformation

Methods	Formula
Pareto scaling	$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{\sqrt{s_i}}$
Vast scaling	$\tilde{x}_{ij} = \frac{(x_{ij} - \bar{x}_i)}{s_i} \cdot \frac{\bar{x}_i}{s_i}$
Level scaling	$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_i}{\bar{x}_i}$

Shifting, Scaling, Standardization, Normalization and Transformation

Methods	Formula
Log transformation	$\tilde{x}_{ij} = \log_{10}(x_{ij})$
Power transformation	$\tilde{x}_{ij} = \sqrt[n]{x_{ij}}$
Box Cox transformation	$\tilde{x}_{ij} = \frac{(x_{ij})^{\delta} - 1}{\delta}$

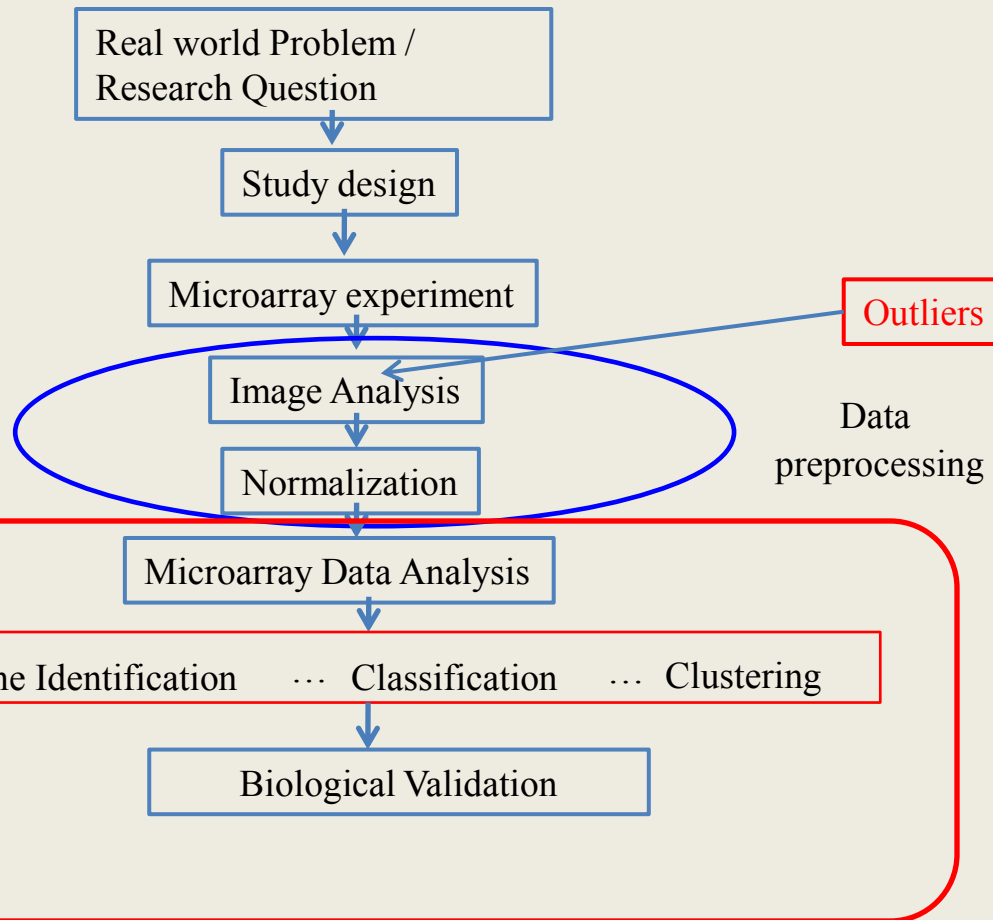
Pipeline of Microarray Gene Expression Analysis



Sources of Outliers

- 1) Human error, such as errors in data collection, recording, or entry.
- 2) Low quality measurements, malfunctioning equipment
- 3) Correct but exceptional data
- 4) Overlapping signals
- 5) due to several steps involves in the data generating processes

Why do We Need to Treatment Outliers?



R-code

```
set.seed(1234)
gene1Cont<-round(rnorm(8,4,1),1)
gene1Cont
gene1Cancer<-round(rnorm(8,6,1))
gene1Cancer
t.test(gene1Cont,gene1Cancer)

gene1Cont[5]<-20

t.test(gene1Cont,gene1Cancer)
```

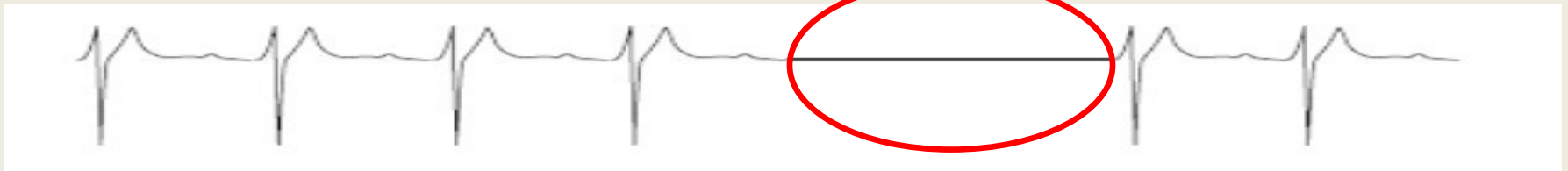
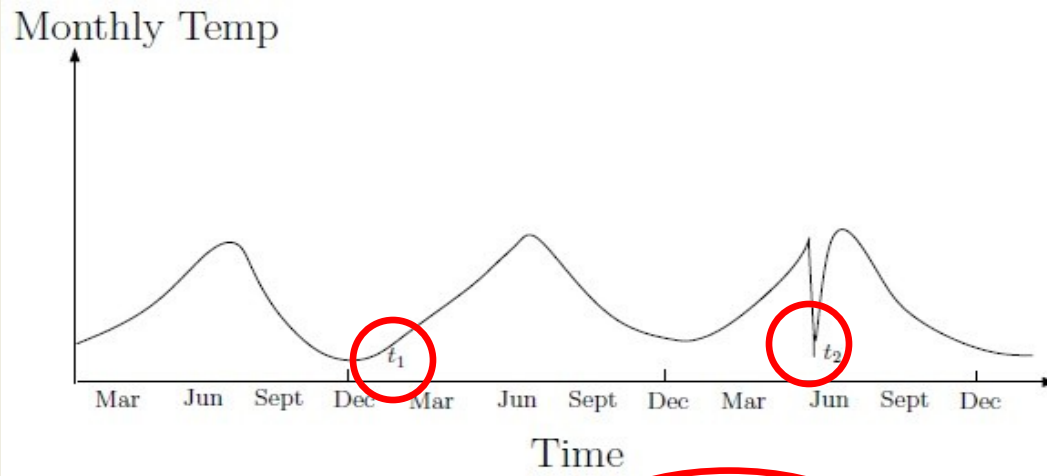
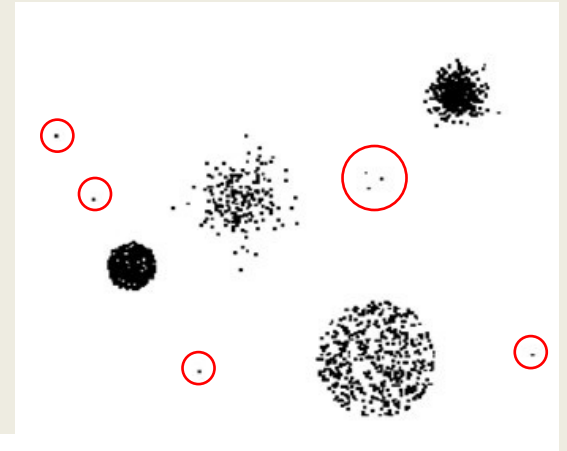
Downstream analysis will be affected by Outliers.
Therefore, treatment of outliers is very important.

Only one outliers can affect
our analysis

Types of Outliers

There are three types of outliers

1. Global outliers
2. contextual / conditional Outliers
3. Collective Outliers



Challenges of Outlier Detection

- Modeling normal objects and outliers properly
 - Hard to enumerate all possible normal behaviors in an application
 - The border between normal and outlier objects is often ill-defined (not having a clear description or limits)
- Application-specific outlier detection
 - Choice of distance measure among objects and the model of relationship among objects are often application-dependent
 - E.g., clinic data: a small deviation could be an outlier; while in marketing analysis, larger fluctuations
- Understandability
 - Understand why these are outliers: Justification of the detection
 - Specify the degree of an outlier: the unlikelihood of the object being generated by a normal mechanism

Outliers Handling Techniques

There are mainly three major techniques for outliers handling

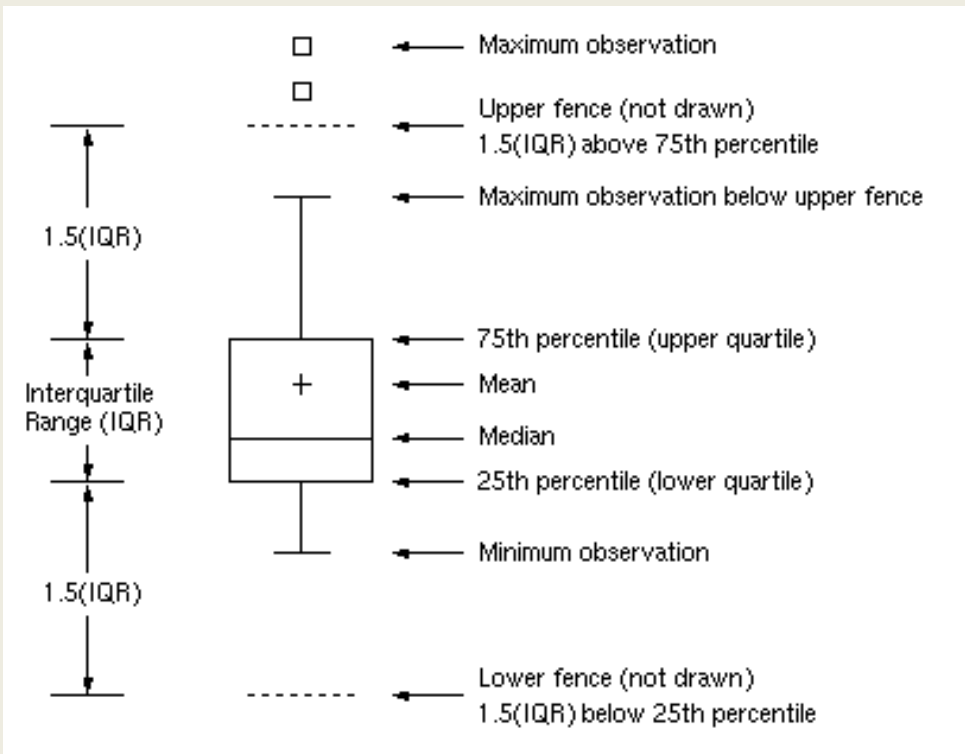
1. Outliers Detection and Deletion
2. Outliers Detection and Modification/Imputation
3. Use of Robust function for parameter estimation

Types of Outliers Detection Schemes

- Graphical
- Model-based (statistical based)
- Distance-based (proximity based)
- Clustering-based

Graphical Outlier Detection

Univariate Approach: Box-and-Whisker Plot



John W. Tukey (1969)

Outliers will be any points

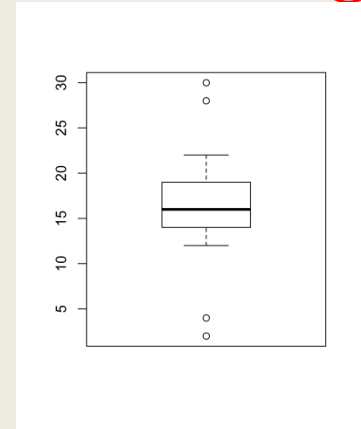
Below $Q_1 - 1.5 \times IQR$

or above $Q_3 + 1.5 \times IQR$

R-code

```
x<-c(15,16,12,12,15,16,14,30,14,17,18,22,20,19,18,2,4,28)
boxplot(x)
```

```
q<-quantile(x)
q
quartile<-q[-1]
quartile
iqr<-quartile[3]-quartile[1]
outObs<-which(x>quartile[3]+1.5*iqr | x<quartile[1]-1.5*iqr)
outObs
outVal<-x[outObs]
outVal
```



Tibshirani and Hastie (2007) suggested ,Outliers will be any points, Below $Q_1 - IQR$ or above $Q_3 + IQR$

Graphical Outlier Detection (Cont.)

The 3-Sigma rule might be the best-known criterion to detect an outlier

Index Plot: 3 Sigma Rule

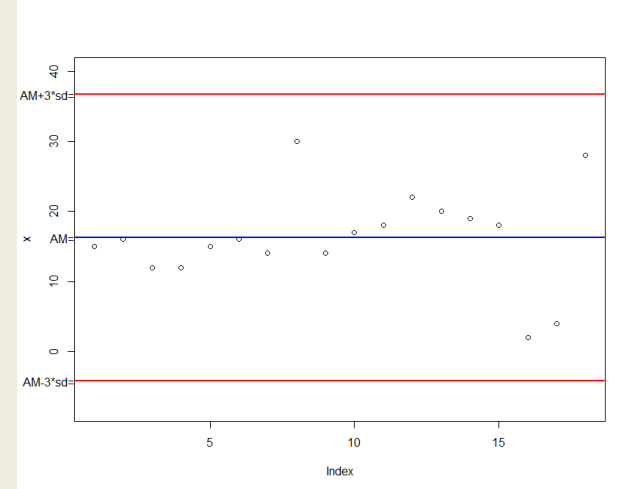
```
x<-c(15,16,12,12,15,16,14,30,14,17,18,22,20,19,18,2,4,28)
am<-mean(x)
sd<-sqrt(var(x))
```

```
plot(x,ylim=c(-8,40))
abline(h = am+3*sd,col="red",lwd=2,)
mtext(paste("AM+3*sd="),side=2,las=1,at=am+3*sd)
```

```
abline(h = am,col="blue",lwd=2,)
mtext(paste("AM="),side=2,las=1,at=am)
```

```
abline(h = am-3*sd,col="red",lwd=2,)
mtext(paste("AM-3*sd="),side=2,las=1,at=am-3*sd)
```

Outliers will be any point
Below $\text{Mean} - 3 \times \text{sd}$
or above $\text{Mean} + 3 \times \text{sd}$



It is likely to have poor performance in practice (Lehman)

Graphical Outlier Detection (Cont.)

Hampel Identifier:

Outliers will be any point
Below Median - $k \times \text{MAD}$
or above Median + $k \times \text{MAD}$

$$\text{MAD} = \frac{1}{0.6745} \text{median} |x_i - \text{median}(x_i)|$$

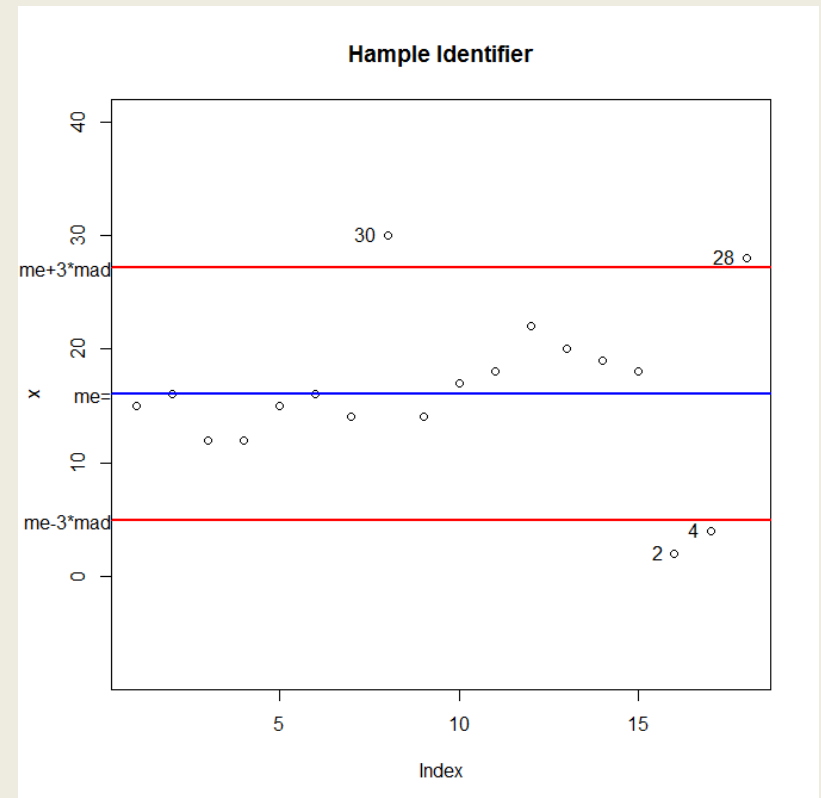
The value of k is usually 3

```
me<-median(x)
md<-mad(x)

plot(x,ylim=c(-8,40))
abline(h = c(me+3*md,me,me-3*md),col=c("red","blue","red"),lwd=2)
mtext(paste("me+3*md"),side=2,las=1,at=me+3*md)
mtext(paste("me="),side=2,las=1,at=me)
mtext(paste("me-3*md"),side=2,las=1,at=me-3*md)
title("Hampel Identifier")

text(which(x>me+3*md | x<me-3*md),x[which(x>me+3*md | x<me-3*md)],labels=outValHi, cex= 1,pos=2)

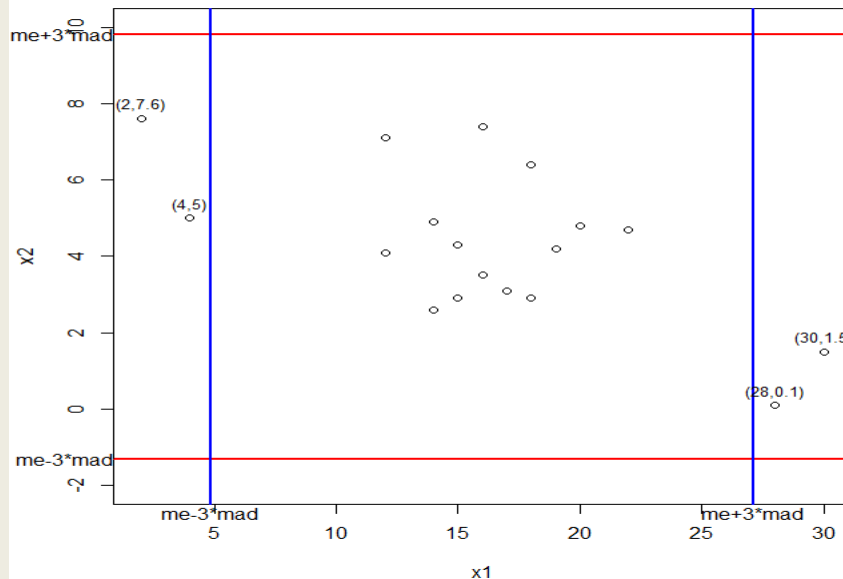
outObsHi<-which(x>me+3*md | x<me-3*md)
outObsHi
outValHi<-x[outObsHi]
outValHi
```



Graphical Outlier Detection (Cont.)

Outlier detection by scatter plot using Hampel identifier

```
x1<-x
x1
set.seed(123)
x2<-round(rnorm(18,4,2),1)
x2
plot(x1,x2,ylim=c(-2,10))
abline(h = c(median(x2)+3*mad(x2),median(x2)-3*mad(x2)),col=c("red","red"),lwd=2)
abline(v = c(median(x1)+3*mad(x1),median(x1)-3*mad(x1)),col=c("blue","blue"),lwd=2)
mtext(paste(c("me+3*mad","me-3*mad")),side=2,las=1,at=c(median(x2)+3*mad(x2),median(x2)-3*mad(x2)))
mtext(paste(c("me+3*mad","me-3*mad")),side=1,las=1,at=c(median(x1)+3*mad(x1),median(x1)-3*mad(x1)))
obs<-which(x1>median(x1)+3*mad(x1) | x1<median(x1)-3*mad(x1)|x2>median(x2)+3*mad(x2) | x2<median(x2)-3*mad(x2))
text(x1[obs],x2[obs],labels=paste('(',paste(x1[obs],x2[obs],sep=','),')',sep="), cex= .8,pos=3)
```



Graphical Outlier Detection (Cont.)

High dimensional data

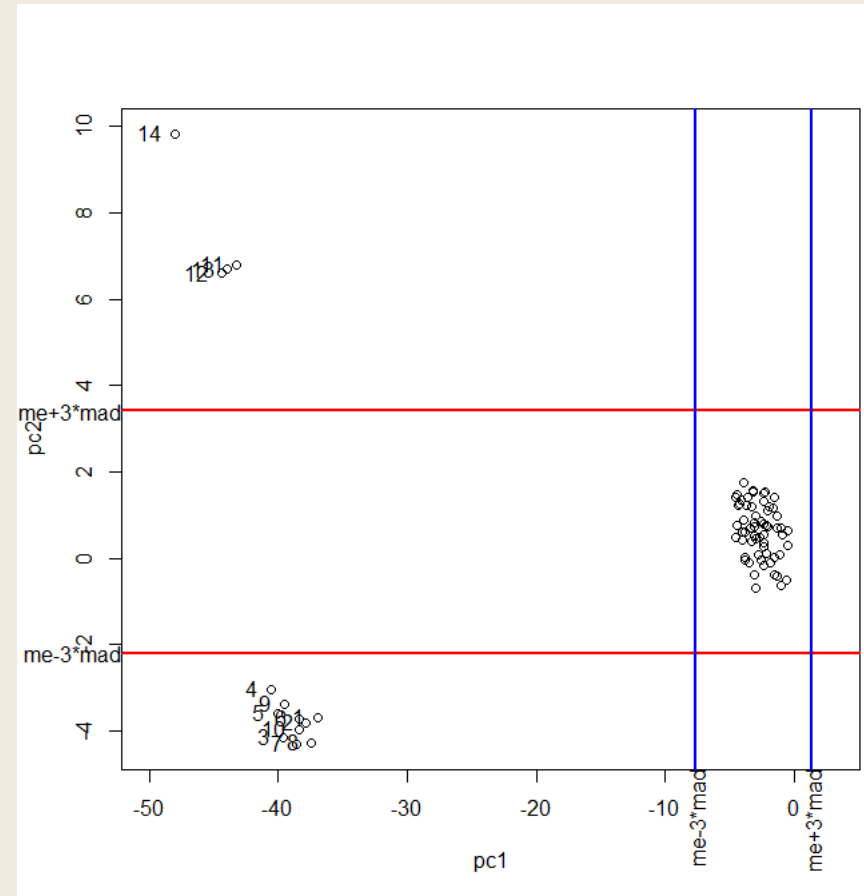
Nishith & Nasser (2012)

Hawkins, Bradu and Kass data (1984)

The first 14 observations are outliers

```
library(wle)
data(artificial)
sv<-svd(artificial)
pc<-sv$u%*%diag(sv$d)
plot(pc[,1],pc[,2],xlim=c(-50,3),xlab='pc1', ylab='pc2')

abline(h = c(median(pc[,2])+3*mad(pc[,2]),median(pc[,2])-
3*mad(pc[,2])),col=c("red","red"),lwd=2)
abline(v = c(median(pc[,1])+3*mad(pc[,1]),median(pc[,1])-
3*mad(pc[,1])),col=c("blue","blue"),lwd=2)
mtext(paste(c("me+3*mad","me-
3*mad")),side=2,las=1,at=c(median(pc[,2])+3*mad(pc[,2]),median(pc[
,2])-3*mad(pc[,2])))
mtext(paste(c("me+3*mad","me-
3*mad")),side=1,las=2,at=c(median(pc[,1])+3*mad(pc[,1]),median(pc[
,1])-3*mad(pc[,1])))
obs<-which(pc[,1]>median(pc[,1])+3*mad(pc[,1]) |
pc[,1]<median(pc[,1])-3*mad(pc[,1])|
pc[,2]>median(pc[,2])+3*mad(pc[,2]) | pc[,2]<median(pc[,2])-
3*mad(pc[,2]))
text(pc[,1][obs],pc[,2][obs],labels=obs, cex= 1,pos=2)
obs
```



The L1-L2 Plot (Ali s. Hadi, 2011)

High Dimensional Multivariate Outlier Detection

The L1-L2 plot is a very simple two-dimensional projection of the data.

It is a scatter plot of

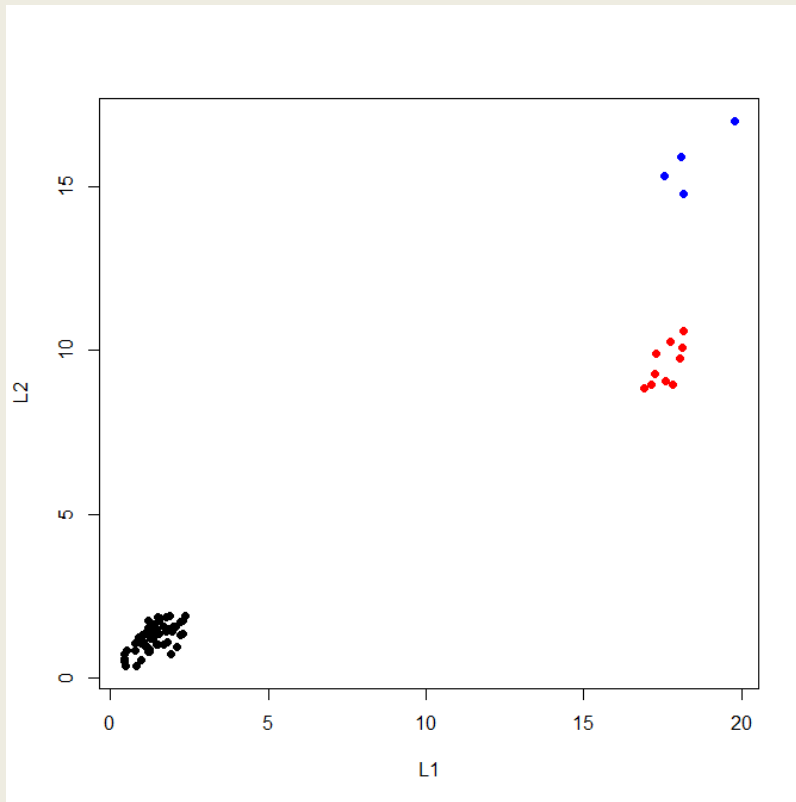
$$L_{1i} = \frac{1}{k} \sum_{j=1}^k |x_{ij}|, \quad i = 1, 2, \dots, n,$$

versus

$$L_{2i} = \left[\frac{1}{k-1} \sum_{j=1}^k (x_{ij} - L_{1i})^2 \right]^{1/2}, \quad i = 1, 2, \dots, n.$$

Graphical Outlier Detection (Cont.)

L1 and L2 plot (Ali s. Hadi, 2011)



Hawkins, Bradu and Kass data (1984)

The first 14 observations are outliers

R-code

```
## L1 calculation function ##
absMean<-function(x){
  average<-sum(abs(x))/length(x)
  return(average)
}
## L2 calculation function ##
absSd<-function(x){
  average<-sum(abs(x))/length(x)
  sd<-sqrt(sum((x-average)^2)/(length(x)-1))
  return(sd)
}

col<-c(rep("red",10),rep("blue",4),rep("black",61))
l1<-apply(artificial,1,absMean)
l2<-apply(artificial,1,absSd)
plot(l1,l2,xlab="L1",pch=16,ylab="L2",col=col)
```

Model Based Outlier Detection

Well known model based outliers detection techniques are

1. Grubb's test (Univariate case)
2. Mahalaobis Distance (Multivariate Case)
3. Regression based outliers detection
4. Principal Component based outlier detection
5. EM algorithm based outlier detection.
6. Classification based outlier detection

Grubbs' Test

- Detect outliers from Univariate data
- Assume data comes from normal distribution
- Detect one outlier at a time, remove the outlier, and repeat

- H_0 : There is no outlier in data

- H_A : There is at least one outlier

- Grubbs' test statistic:

$$G = \frac{\max |X - \bar{X}|}{s}$$

- Reject H_0 if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N, N-2)}}{N-2 + t^2_{(\alpha/N, N-2)}}}$$

Grubbs' Test

R-Code for Grubbs' Test

```
library(outliers)
x<-c(15,16,14,14,15,16,16,40,15,17,18,22,20,19,18,1,2,58)
grubbs.flag <- function(x) {
  outliers <- NULL
  test <- x
  grubbs.result <- grubbs.test(test)
  pv <- grubbs.result$p.value
  while(pv < 0.05) {
    outliers <- c(outliers,as.numeric(strsplit(grubbs.result$alternative," ")[[1]][3]))
    test <- x[!x %in% outliers]
    grubbs.result <- grubbs.test(test)
    pv <- grubbs.result$p.value
  }
  return(data.frame(X=x,Outlier=(x %in% outliers)))
}
```

```
grubbs.flag(x)
```

```
> grubbs.flag(x)
```

	X	Outlier
1	15	FALSE
2	16	FALSE
3	14	FALSE
4	14	FALSE
5	15	FALSE
6	16	FALSE
7	16	FALSE
8	40	TRUE
9	15	FALSE
10	17	FALSE
11	18	FALSE
12	22	FALSE
13	20	FALSE
14	19	FALSE
15	18	FALSE
16	1	TRUE
17	2	TRUE
18	58	TRUE

Mahalanobis Distance Based Outlier Detection

Compute squared Mahalaobis distance (Multivariate Case)

Let $\bar{X}$ be the mean vector for a multivariate data set. Mahalaobis distance for an object X to $\bar{X}$ is $Mdist(X, \bar{X}) = (X - \bar{X})^T S^{-1} (X - \bar{X})$ where S is the covariance matrix.

Assume that X follows multivariate normal distribution

$MDist$ follows a χ^2 -distribution with d degrees of freedom (d = data dimensionality)

Apply this method in Hawkins, Bradu and Kass data (1984)

R-code

```
sMD<-mahalanobis(artificial, colMeans(artificial), cov(artificial))  
outlierObs<-which(sMD>qchisq(0.975, df=4))  
outlierObs
```

```
> outlierObs  
[1] 11 12 13 14
```

Robust Mahalanobis Distance

Compute squared robust Mahalanobis distance (Multivariate Case) using the **robust location and robust covariance matrix**.

Assume that X follows multivariate normal distribution

$MDist$ follows a χ^2 -distribution with d degrees of freedom (d = data dimensionality)

Apply this method in Hawkins, Bradu and Kass data (1984)

R-code

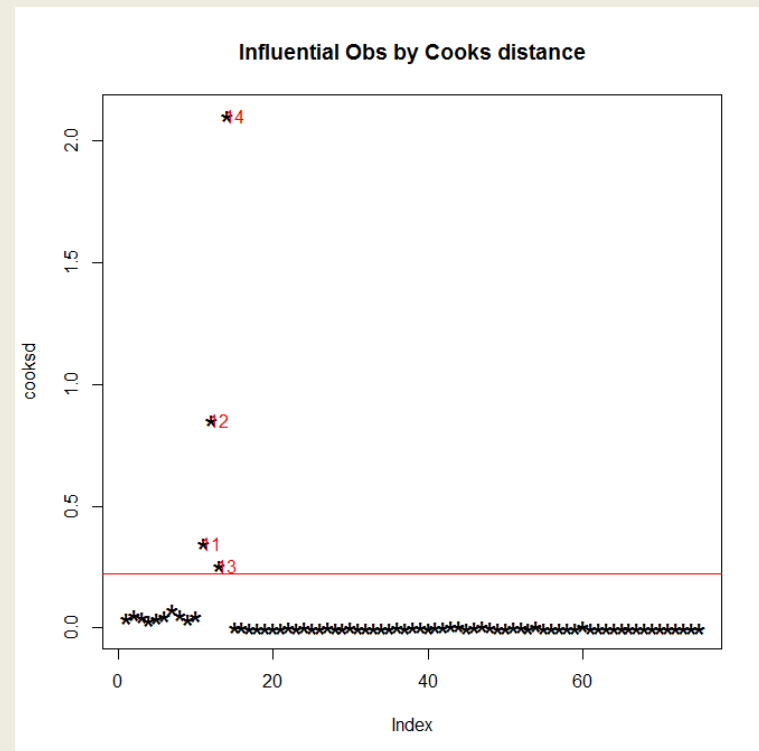
```
library(robustbase)
ld<-covMcd(artificial)
rsMD<-mahalanobis(artificial,ld$center , ld$cov)
outliersObs<-which(rsMD>qchisq(0.975, df=4))
outliersObs
```

```
> outliersObs
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

Regression Based Influential Observations Detection Using Cooks Distance

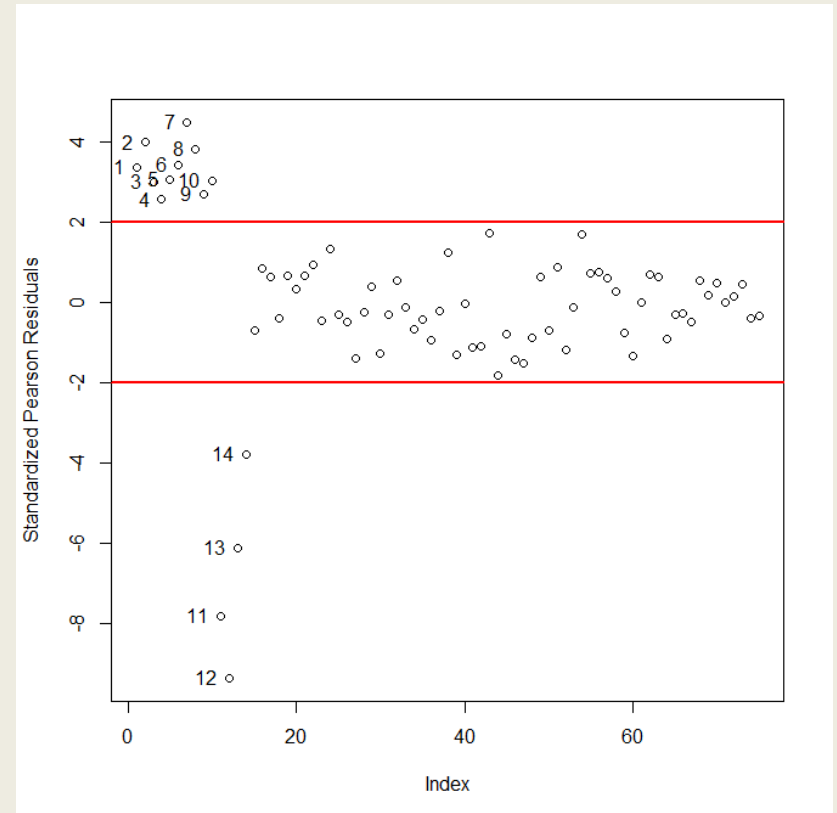
```
linReg<-lm(y~.,data=artificial)
cooksds<-cooks.distance(linReg)[1:14]
plot(cooksds, pch="*", cex=2, main="Influential Obs by Cooks distance") # plot cook's distance
abline(h = 4*mean(cooksds, na.rm=T), col="red") # add cutoff line
text(x=1:length(cooksds)+1, y=cooksds, labels=ifelse(cooksds>4*mean(cooksds,
na.rm=T),names(cooksds),""), col="red")
```



Regression Based Outlier Detection Using Standardized Pearson Residuals

R-Code

```
linReg<-lm(y~.,data=artificial)
resid<-residuals(linReg, type="pearson")
plot(residuals(linReg, type="pearson"),ylab="Standardized Pearson Residuals")
abline(h=c(2,-2), col=c("red" ,"red"),lwd=2)
obs<-which(resid>2 | resid<(-2))
text(obs,resid[obs],labels=obs, cex= 1,pos=2)
obs
```



EM Algorithm Based Outlier Detection

Robust Model-based Learning via Spatial-EM Algorithm(Yu et. al., 2015)

```
library(RobustEM)
cluster_em_outlier(iris[-5],3,"rcm")$clusters
```

```
> cluster_em_outlier(iris[-5],3,"rcm")$clusters
 [1] "3"      "3"      "3"      "3"      "3"      "3"      "3"
 [8] "3"      "3"      "3"      "3"      "3"      "3"      "3"
[15] "3"      "3"      "3"      "3"      "3"      "3"      "3"
[22] "3"      "outlier" "3"      "3"      "3"      "3"      "3"
[29] "3"      "3"      "3"      "3"      "3"      "3"      "3"
[36] "3"      "3"      "3"      "3"      "3"      "3"      "outlier"
[43] "3"      "outlier" "3"      "3"      "3"      "3"      "3"
[50] "3"      "1"      "1"      "1"      "1"      "1"      "1"
[57] "1"      "1"      "1"      "1"      "1"      "1"      "1"
[64] "1"      "1"      "1"      "1"      "1"      "2"      "1"
[71] "1"      "1"      "2"      "1"      "1"      "1"      "1"
[78] "1"      "1"      "1"      "1"      "1"      "1"      "1"
[85] "1"      "1"      "1"      "1"      "1"      "1"      "1"
[92] "1"      "1"      "1"      "1"      "1"      "1"      "1"
[99] "1"      "1"      "2"      "2"      "2"      "2"      "2"
[106] "2"      "1"      "2"      "2"      "2"      "2"      "2"
[113] "2"      "2"      "2"      "2"      "2"      "2"      "outlier"
[120] "2"      "2"      "2"      "2"      "2"      "2"      "2"
[127] "2"      "1"      "2"      "2"      "2"      "outlier" "2"
[134] "1"      "outlier" "2"      "2"      "1"      "1"      "2"
[141] "2"      "2"      "2"      "2"      "2"      "2"      "2"
[148] "2"      "2"      "1"      "1"      "1"      "1"      "1"
> |
```

Principal Component Analysis for Outlier Detection

OutlierPCDist : Outlier identification in high dimensions using the PCDIST algorithm

A.D. Shieh and Y.S. Hung (2009), Detecting Outlier Samples in Microarray Data, Statistical Applications in Genetics and Molecular Biology Vol. 8.

M. Zhu, and A. Ghodsi (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. Computational Statistics & Data Analysis, Vol. 51, 918-930.

R-Code

```
library(rrcovHD)  
OutlierPCDist(iris[-5])
```

Call:

```
OutlierPCDist.default(x = iris[-5])
```

-> Method: PCDIST

```
OutlierPCDist(artificial)
```

Number of outliers detected: 15

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 34
```

Number of outliers detected: 7

```
[1] 42 115 118 132 135 142 146
```

Principal Component Analysis for Outlier Detection

OutlierPCOut: Outlier identification in high dimensions using the PCOUT algorithm

P. Filzmoser, R. Maronna and M. Werner (2008), Outlier identification in high dimensions, Computational Statistics & Data Analysis, Vol. 52 1694–1711.

R-Code

```
library(rrcovHD)  
OutlierPCDist(iris[-5])
```

Call:

```
OutlierPCOut.default(x = iris[-5], explvar = 0.99, trace = FALSE)  
-> Method: PCOUT
```

Number of outliers detected: 15

```
[1] 15 16 42 61 63 88 101 107 115 118 122 123 132 137 149
```

```
OutlierPCOut(artificial, explvar=0.99, trace=FALSE)
```

Number of outliers detected: 15

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 34
```

Cluster Based Outlier Detection

A unified approach to clustering and outlier detection(Chawla and Gionis, 2013)
SIAM International Conference on Data Mining (SDM13)

Cluster Based Outlier Detection Algorithm For Healthcare Data (Christy et. Al. 2015)

Computational process:

- ❖ Compute the cluster center for each cluster
- ❖ Determine the Euclidean distance between each observation and corresponding cluster center
- ❖ Arrange all the distance in ascending order
- ❖ Declare specific portion of observation as outlier according to the distances

Cluster Based Outlier Detection

R-code

```
# remove species from the data to cluster
iris2 <- iris[,1:4]
kmeans.result <- kmeans(iris2, centers=3)
# cluster centers
kmeans.result$centers
# calculate distances between objects and cluster centers
centers <- kmeans.result$centers[kmeans.result$cluster, ]
distances <- sqrt(rowSums((iris2 - centers)^2))
# pick top 5 largest distances
outliers <- order(distances, decreasing=T)[1:5]
# who are outliers
print(outliers)

> print(outliers)
[1] 119 118 132 123 106 61 136
```

Using Cluster Median

```
library(matrixStats)
iris2 <- iris[,1:4]
kmeans.result <- kmeans(iris2, centers=3)

kmeans.result$cluster

c1<-colMedians(as.matrix(iris2[which(kmeans.result$cluster==1),]))
c2<-colMedians(as.matrix(iris2[which(kmeans.result$cluster==2),]))
c3<-colMedians(as.matrix(iris2[which(kmeans.result$cluster==3),]))
clusterCenter<-rbind(c1,c2,c3)

# cluster centers

clusterCenter

# calculate distances between objects and cluster centers
centers <-clusterCenter[kmeans.result$cluster, ]
distances <- rowSums((iris2 - centers)^2)
order(distances, decreasing=T)[1:5]

> order(distances, decreasing=T)[1:5]
[1] 119 118 132 61 123 106 94
```

Depth Based Outlier Detection

Depth: The depth of a point relative to a given data set measures how deep that point lies in the data cloud.

Spatial Depth: The spatial outlyingness is given by $O_{SP}(x, F) = \|ES(x - \mathbf{X})\|$.

Where, X has distribution F and

$$S(x) = \begin{cases} \frac{x}{\|x\|}, & \text{if } x \neq 0, \\ 0, & \text{if } x = 0, \end{cases}$$

The Corresponding Spatial Depth $D_{SP}(x, F) = 1 - \|ES(x - \mathbf{X})\|$

Spatial depth based outlier detection is proposed by Chen - 2009

R-Code

```
library(depth.plot)
spatial.outlier(artificial, x = artificial, threshold = 0.2)

$index
[1] 3 4 5 7 9 11 12 13 14
```

Kernel Depth Based Outlier Detection

ISSN 2320-5407

International Journal of Advanced Research (2016), Volume 4, Issue 1, 416- 425

Proposed Kernel based Depth

In this article we have used our new proposed kernel based depth technique (using a radial basis kernel function) to visualize multivariate data.

$$\text{Kernel Depth} = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^n K(x_i, x_j); \quad \text{for } i = 1, 2, \dots, n.$$

```
library(kernlab)
rbf <- rbfdot(sigma = 0.05)
kd <- kernelMatrix(rbf, as.matrix(artificial))
skd <- (rowSums(kd) - diag(kd)) / 74
outlier <- which(skd < 0.15)
outlier

> outlier
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

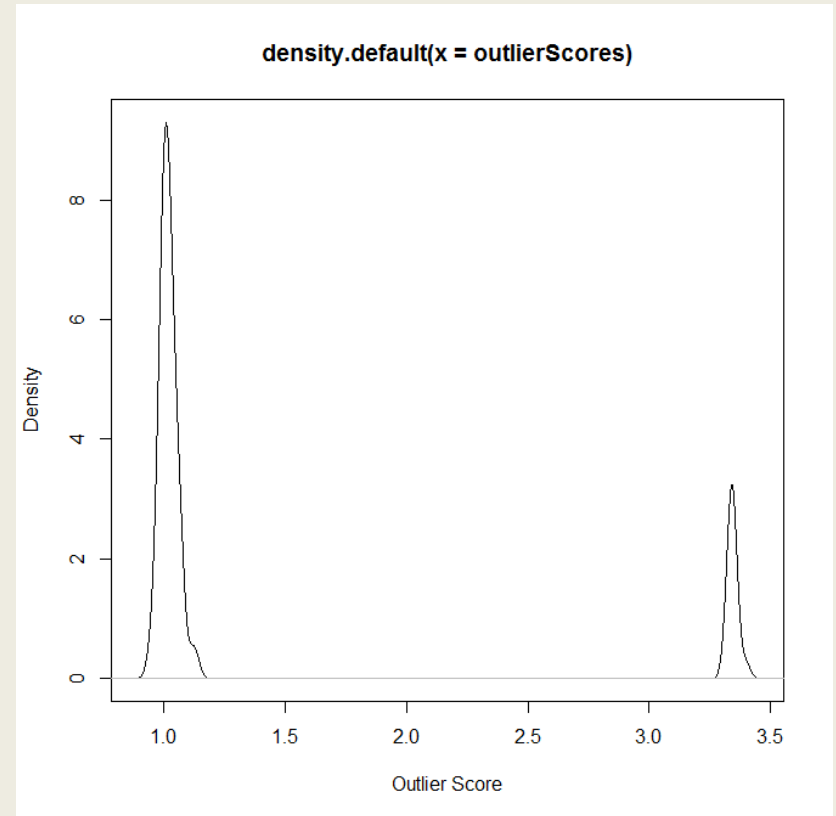

Local Outlier Factor Algorithm

LOF (Local Outlier Factor) is an algorithm for identifying density-based local outliers [Breunig et al., 2000].

```
library(Rlof)
outlierScores <- lof(artificial, k=15)
outlierScores
plot(density(outlierScores), xlab="Outlier Score")

outliers <- which(outlierScores > 3)
outliers

> outliers
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```



Classification Based Outliers Detection

Residuals and regression diagnostics: focusing on logistic regression

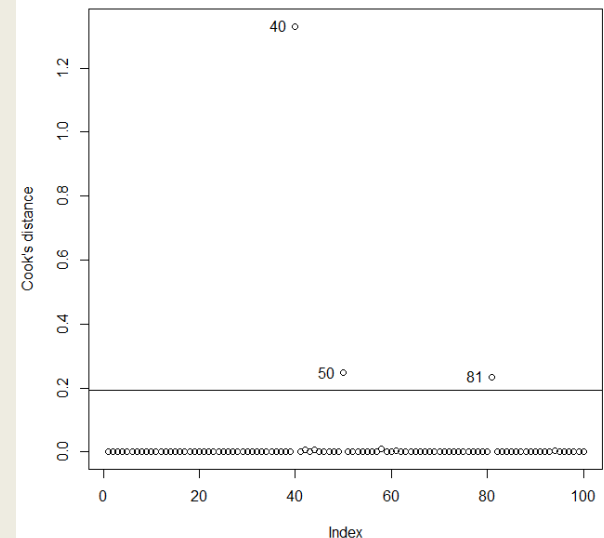
Zhongheng Zhang

Department of Critical Care Medicine, Jinhua Municipal Central Hospital, Jinhua Hospital of Zhejiang University, Jinhua 321000, China

Correspondence to: Zhongheng Zhang, MMed. 351#, Mingyue Road, Jinhua 321000, China. Email: zh_zhang1984@hotmail.com.

Logistic Regression Based Outlier Detection

```
ir<-iris[1:100,]  
cl<-c(rep(0,49),rep(1,31),0,rep(1,19))  
irm<-cbind(ir[, -5], cl)  
irm[40,3]<-30  
  
glm.out = glm(cl ~., family=binomial(logit), data=irm)  
plot(cooks.distance(glm.out), ylab="Cook's distance")  
abline(h=mean(influence(glm.out)$hat)*qchisq(0.95, df=1))  
obs<-which(cooks.distance(glm.out)>mean(influence(glm.out)$hat)*qchisq(0.95,  
df=1))  
text(obs, cooks.distance(glm.out)[obs], labels=obs, cex= 1, pos=2)
```



Missing Value Imputation Technique

- 1. Missing Value Imputation Using Mean**
- 2. Missing Value Imputation Using Median**
- 3. Missing Value Imputation Using Half of the Minimum Value**
- 4. Imputation of Missing Values Using Zero**
- 5. Imputation of Missing Values Using k-Nearest Neighbours (kNN)**
- 6. Imputation of Missing Values Using Random Forest**
- 7. Imputation of Missing Values Using Expectation Maximization Algorithm**

Missing Value Imputation Technique

Missing Value Imputation Using Mean, Median and Half of the Minimum Value

Mean and median imputation substitutes every missing value with the mean and median of the non-missing values of the corresponding gene. Half of the minimum value imputation substitutes each missing value with the half of the minimum value of the corresponding gene.

Imputation of Missing Values Using Zero

Zero imputation is that all the missing values in the data matrix are replaced by zero. Sometimes low intensity of a gene produces missing values therefore missing values are imputed by zero values.

Imputation of Missing Values Using k-Nearest Neighbours (kNN)

Let X be a microarray data matrix with $p \times n$, p is the number of genes and n is the number of samples. Let X^c be the complete gene set i.e. all rows are nonmissing and X^m the remainder with at least one missing for each row. Let $x^* \in X^m$. K nearest neighbor algorithm (Hastie et al., 1999; Troyanskaya et al., 2001) for imputing the missing values in x^* :

- i) Compute the Euclidean distance between x^* and all the metabolites in X^c , using only those co-ordinates that are non-missing in x^* . Identify the K closest on the basis of Euclidean distance.
- ii) Impute the missing coordinates of x^* by least square approach using the corresponding coordinates of K closest; where, K is the tuning parameter selected by the researcher or depends on the structure of the data.

However, this can fail if all the neighbors are missing in a particular element. In such a case we can use the overall column mean for that block of genes. In R platform this method can be found in R library “*impute*”.

Imputation of Missing Values Using Random Forest

Random forest is a tree based regression and classification technique that is suitable for both parametric and nonparametric dataset (Breiman, 2001). In 2012, Stekhoven and Bühlmann developed an algorithm for imputing missing values using random forest (Stekhoven and Bühlmann, 2012). The procedure of the missing imputation technique according to Stekhoven and Bühlmann is given below.

Let X is a $n \times p$ data matrix, here the number of rows n represents the number of samples and number of columns i.e., p represents the number of genes. Firstly, for any arbitrary variable (column of X) X_a including missing values with observation number $i_{mis}^a \in \{1, 2, \dots, n\}$ is separated the data into four parts, first part is the observed values of the variable X_a , denoted by y_{obs}^a ; the second part is the missing values of the variable X_a , denoted by y_{mis}^a ; third part is the variables other than X_a with observations $i_{obs}^a = \{1, 2, \dots, n\} \setminus i_{mis}^a$ denoted by x_{obs}^a and the last part is the

Imputing missing values using Random Forest (Cont.)

variables other than X_a with observation i_{mis}^a denoted by x_{mis}^a . Here, x_{obs}^a is typically not completely non-missing, on the other hand x_{mis}^a is typically not completely missing, because, i_{mis}^a and i_{obs}^a are calculated on the basis of X_a . Initially, missing values of X is imputed using mean of the corresponding variable and then, sort the columns of X , X_a ; $a=1,2,\dots,p$ according to the lowest number of missing values. For each variable X_a , fit the random forest y_{obs}^a on x_{obs}^a and predict y_{mis}^a by applying trained random forest to x_{mis}^a . The imputation is continued until the stopping criterion satisfied. In R platform this method can be found in library “*missForest*”.

Imputation of Missing Values Using Expectation Maximization Algorithm

The expectation maximization (EM) algorithm was explained by Arthur Dempster, Nan Laird, and Donald Rubin in 1977. Suppose $X_1, X_2, \dots, X_n$ be a sample from $N_p \sim (\mu, \Sigma)$. They wanted to make inference on $\Theta = (\mu, \Sigma)$, despite the missing observations. For non-missing dataset the maximum likelihood estimate (MLE) is obtained by choosing Θ for which

$\ell(\Theta) = -\frac{n}{2} \log \Sigma - \frac{1}{2} \sum_{i=1}^n (X_i - \mu)^T \Sigma^{-1} (X_i - \mu) + c$ is maximized. For missing data, EM algorithm

provides an iterative scheme with current values of Θ , replace the missing values by their conditional expectations given the observed data and the current guesses for Θ . The algorithm of the above technique is given below,

Imputation of missing values using Expectation maximization Algorithm (Cont.)

- i) Let X_{obs} be the observed data, X_{mis} be the missing data and $X = (X_{mis}, X_{obs})$ be the complete data.
- ii) The EM algorithm consists of expectation step and maximization step. The expectation step calculates the expectation of the complete likelihood for given observations X_{obs} .
i.e., let $\Theta^{(i)}$ be the current guess for Θ and the calculation
$$Q(\Theta, \Theta^{(i)}) = \int \ell(\Theta; X_{mis}, X_{obs}) f(X_{mis} | X_{obs}; \Theta^{(i)}) dX_{mis}.$$
 The maximization step maximizes $Q(\Theta, \Theta^{(i)})$ as a function of Θ to get $\Theta^{(i+1)}$.
- iii) Iterate till convergence to get maximum likelihood estimate $\hat{\Theta}$.

Treatment of Outlier

The most straightforward way is to delete the observation/sample whenever a outlier is present . Another way of handling outliers is whether modified the outlier observation or use the robust function for further downstream analysis.

In case of outlier modification, usually outlier cells are considered as missing values and impute those using different missing value imputation techniques.

Missing Value Imputation Methods

Mean, median, kNN, random forest, half of the minimum value found in each metabolite, Probabilistic principal component analysis (PPCA), Bayesian PCA (BPCA), multiple imputation with expectation maximization (EM) algorithm, monte carlo markov chain (MCMC) method, zero and SVD.

Among these techniques random forest imputation is better performer
Gromski et al. (2014)



Identification of Differentially Expressed Gene Using Robust Singular Value Decomposition

Authors: Kumar, Nishith; Tofazzal Hossain, Md.; J. Beh, Eric; Sugimoto, Masahiro; Nasser, Mohammed

Source: *Current Bioinformatics*, Volume 11, Number 3, July 2016, pp. 366-374(9)

Publisher: [Bentham Science Publishers](#)

Buy Article:

\$63.00 plus tax

[\(Refund Policy\)](#)

[ADD TO CART](#)

[BUY NOW](#)

[< previous article](#) | [view table of contents](#) | [next article >](#)

[♥ ADD TO FAVOURITES](#)

Abstract:

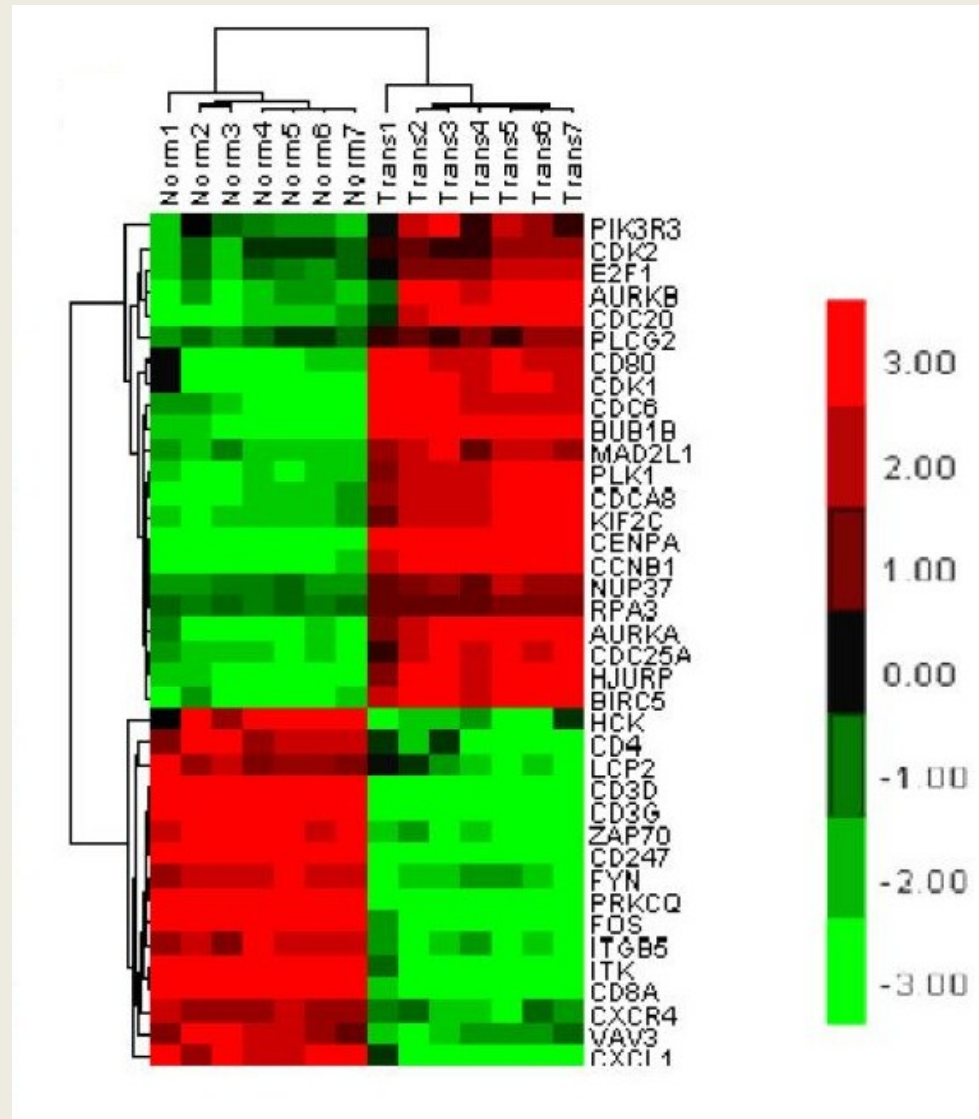
Identification of differentially expressed genes (DEG) in transcriptomic analyses is one of the important tasks to find out significantly activated/deactivated pathways. Outliers and/or the missing values are commonly observed in microarray data; however, most available statistical methods did not deal with these issues and, therefore, their analytical results were frequently skewed and deteriorated. Here, we developed a novel technique robust against outliers and missing values: a dimension reduction procedure based on robust singular value decomposition (RSVD). The RSVD was evaluated by two numerical experiments: artificially prepared and nonsmall cell lung cancer data (gene expression data). Four conventional techniques, such as Student's t-test, SAM, Bayesian Robust Inference for Differential Gene Expression (BRIDGE) and Linear models for microarray data (Limma), were also performed. We evaluated the area under curve (AUC) from receiver operating characteristic curves of these five

Differential Expression of Gene

The gene which can differentiate biological samples into two or more biological conditions is called the differentially expressed gene and its expression values are called differential expressions. For example- the clinical outcomes are linked with some particular genes or class of genes significantly, through a regression model then we say that the genes or gene sets could be “differentially expressed”.

Genes	C1	C2	C3	T1	T2	T3	P-value
G8522	6.78	6.55	6.37	6.89	6.78	6.92	1.000 (EE)
G8523	6.52	6.61	6.72	6.51	6.59	6.46	0.850 (EE)
G8524	5.67	5.69	5.88	7.43	7.16	7.31	0.001 (DE)
G8525	5.64	5.91	5.61	7.41	7.49	7.41	0.005 (DE)

Differential Expression of Gene



Differentially Expressed Gene Identification Methods

Four types of statistical procedure have primarily been used to identify differential metabolites:

- (i) classical parametric approaches, such as Student's t -test, classical volcano plot (CVP) and fold change rank ordering statistics (FCROS)
- (ii) classical non-parametric approaches, such as significance analysis of microarrays (SAM), and the Wilcoxon and Kruskal-Wallis (KW) tests
- (iii) Bayesian parametric approaches, such as Bayesian robust inference for differential gene expression (BRIDGE), empirical Bayes methods for microarrays (EBarrays), and linear models for microarrays (Limma), and
- (iv) Bayesian non-parametric approaches.

Differentially Expressed Gene Identification Methods

1. Calculation of fold change (FC) value

Probe ID	Disease	Control
1007_s_at	35 40 27 24 26 35 22 40 35 45	12 15 16 14 11 12 14 15 13 14

$$\begin{aligned}\text{Fold change (FC)} &= (\text{Mean of } x_1 / \text{Mean of } x_2) \\ &= 2.41\end{aligned}$$

2. Calculation of p -value using t -test

Probe ID	Disease	Control
1007_s_at	18 15 18 16 17 18 15 14 18 17	12 15 16 14 11 12 14 15 13 14

We want to test the hypothesis :

$$H_0: \mu_1 = \mu_2$$

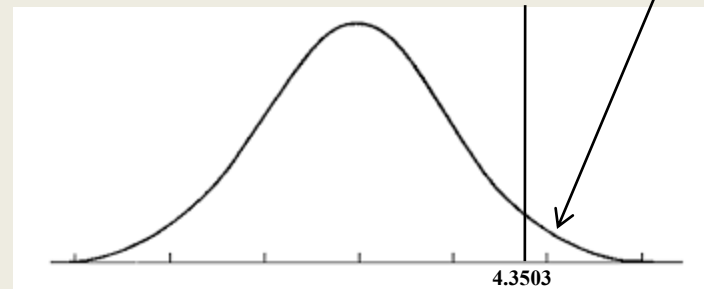
$$H_a: \mu_1 \neq \mu_2$$

Test Statistic:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}$$

$$s^2 = \frac{\sum_{i=1}^{n_1} (x_i - \bar{x}_1)^2 + \sum_{j=1}^{n_2} (x_j - \bar{x}_2)^2}{n_1 + n_2 - 2}$$

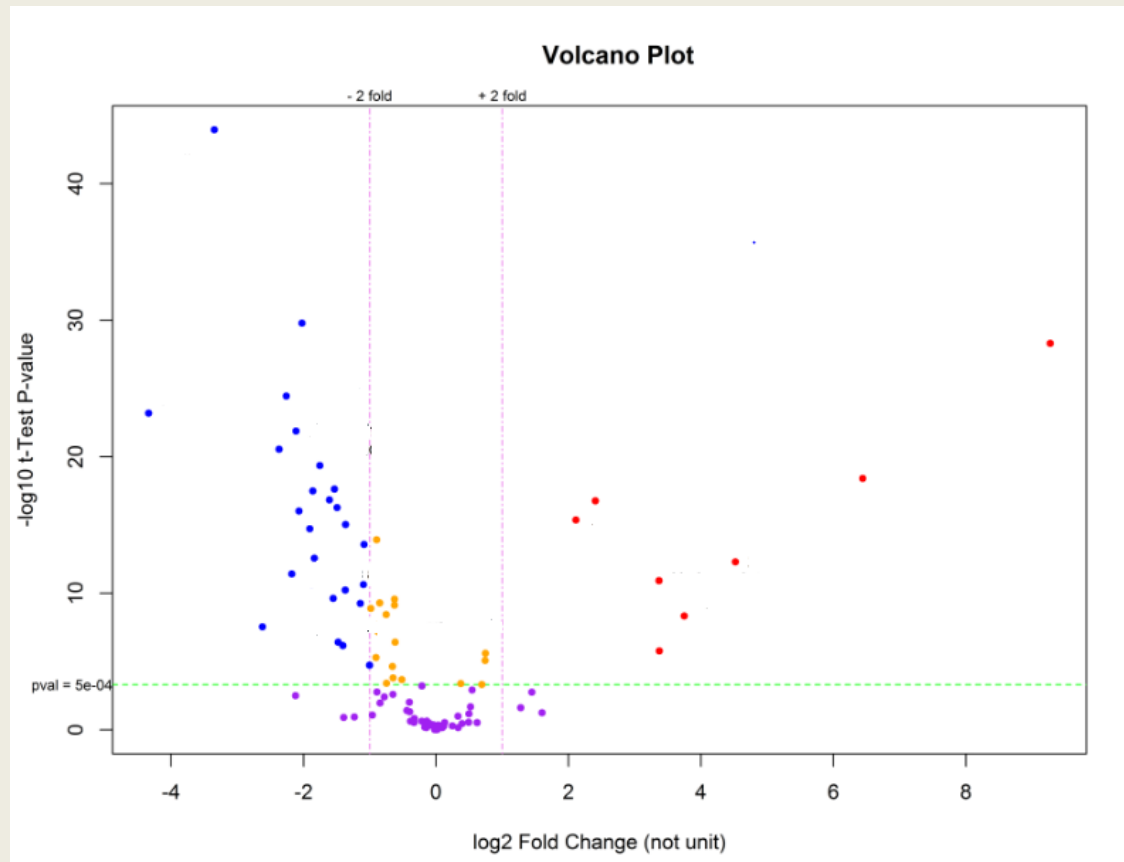
Calculated $t = 4.3503$



3. Classical Volcano Plot

1. Calculate fold change (FC) value
2. Calculate p-value

plot $\log_2$ (fold-change) on the X-axis against $-\log_{10}$ (p-value) from the t-test on the Y-axis.



4. Significance Analysis of Microarray (SAM)

Tusher, Tibshirani and Chu (2001) proposed the significance analysis of microarrays' (SAM) for finding differentially expressed genes.

The SAM statistic is defined as follows

$$d(i) = \frac{\bar{X}_i - \bar{Y}_i}{s(i) + s_0}$$

where s_0 is the percentile of the distribution of s , this constant term is chosen to minimize the coefficient of variation of $d(i)$. $s(i)$ is the gene-specific scatter, which is defined as

$$s(i) = \sqrt{a \left\{ \sum_{j=1}^J (X_{ij} - \bar{X}_i)^2 + \sum_{k=1}^K (Y_{ik} - \bar{Y}_i)^2 \right\}}$$

Where $a = (1/j + 1/k) / (j + k - 2)$.

5. Kruskal Wallis Test

This is a non-parametric test was proposed by Kruskal and Wallis and it is used when the data does not satisfy the normality property and contains outliers. The test statistic of Kruskal-Wallis for k groups each of size n_i is defined by

$$T = \frac{1}{s^2} \left[\sum_{i=1}^k \frac{R_i^2}{n_i} - N \frac{(N+1)^2}{4} \right]$$

where, N is the total number and R_i is the sum of the ranks for the i -th sample and

$$s^2 = \frac{1}{N-1} \left[\sum_{i,j} R_{ij}^2 - N \frac{(N+1)^2}{4} \right]$$

we assume H_0 is that all k distribution functions are equal.

6. ANOVA

Let x_{jk} be the k th observed random expression of a gene in the j th condition ($j = 1, 2, \dots, m$; $k = 1, 2, \dots, n_j$), which follows the one-way ANOVA model as expressed below:

$$x_{jk} = \mu_j + \epsilon_{jk},$$

where μ_j is the mean of all expressions of a gene in the j th condition and ϵ_{jk} is the random error term that follows $N(0, \sigma_j^2)$. We wish to test the null hypothesis (H_0): $\mu_1 = \mu_2 = \dots = \mu_m = \mu$ against the alternative hypothesis (H_1): H_0 is not true, assuming that $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_m^2 = \sigma^2$. Thus, the generalized likelihood-ratio test (LRT) criterion yields the following F -statistic to test H_0 against H_1 :

$$F = \frac{\sum_{j=1}^m n_j (\hat{\mu}_j - \hat{\mu})^2 / (m - 1)}{[n_1 \hat{\sigma}_1^2 + n_2 \hat{\sigma}_2^2 + \dots + n_m \hat{\sigma}_m^2] / (n - m)},$$

which follows the F -distribution with $(m-1)$ and $(n-m)$ degrees of freedom under H_0 , where $n = n_1 + n_2 + \dots + n_m$ and $\hat{\mu} = \sum_{j=1}^m n_j \hat{\mu}_j / n$. Here, $\hat{\mu}_j$ and $\hat{\sigma}_j^2$ are the maximum likelihood estimates (MLEs) of μ_j and σ_j^2 , respectively, for the j th condition/group.

Bonferroni Correction

When multiple dependent or independent tests are made on a single dataset, Bonferroni correction is needed for adjusting the level of significance. This correction method was developed by Carlo Emilio Bonferroni in 1936. We used the Bonferroni correction to reduce the chances of obtaining false-positive results (type I errors) when multiple tests are conducted on a single set of data. Without the use of Bonferroni correction, then the probability to incorrectly identify significant results will increase.

Bonferroni procedure is used for multiple testing to control the family wise error rate (FWER) at the level α . If $H_1, H_2, \dots, H_m$ be a family of null hypotheses then this procedure reject the null hypothesis, whose unadjusted p-value is less than or equal to α/m , where m denotes the total numbers of genes or null hypotheses.

Confusion matrix for two class prediction problem

Matching Matrix	True Values			Marginal Total
		DE (Positive)	EE (Negative)	
Prediction Outcomes	DE (Positive)	True Positive (TP)	False Positive (FP) (Type I error)	M1
	EE (Negative)	False Negative (FN) (Type II error)	True Negative (TN)	M2
Marginal Total		N1	N2	N

TPR, TNR, FPR, FNR and ROC

TPR (True Positive Rate) = $TP/(TP+FN)$
(Sensitivity)

FPR (False Positive Rate) = $FP/(TN+FP)$

TNR (True Negative Rate) = $TN/(TN+FP)$
(specificity)

FNR (False Negative Rate) = $FN/(TP+FN)$

FDR (False Discovery Rate) = $FP/(FP+TP)$

ROC (Receiver operating characteristic): Plot of **TPR** vs **FPR**

Objectives and Application	Statistical Methods
Data Transformation	Log transformation, Box-cox transformation, Square root transformation, generalized logarithmic transformation etc.
Data Normalization	Auto scaling, mean/median/quantile scaling, Pareto scaling etc.
Missing Value Problem	Mean, median, kNN, random forest, half of the minimum value found in each metabolite, Probabilistic principal component analysis (PPCA), Bayesian PCA (BPCA), multiple imputation with expectation maximization (EM) algorithm, monte carlo markov chain (MCMC) method, zero and SVD.
Outliers Problem	Detection: 3 sigma rule, IQR rule etc. Different Robust techniques: Robust SVD, Robust PCA, Robust regression, Robust Hierarchical Clustering etc.
Dimensionality Problem	PCA,FA, SVD etc.
Differentially Expressed Gene Identification / Feature Selection	Filtering Methods: student's <i>t-test</i> , SAM, FC, ANOVA, MANOVA etc. Wrapper Methods: simulation annealing, genetic algorithm etc. Embedded Methods: Decision tree, naive bayes, SVM etc.
Dose Selection / Personalized Medicine	Nonlinear Mixed Effect Modeling, Nested ANOVA, ASCA etc.
System Biology / Pathway Analysis	GSEA, Bayesian Network etc.

Thank You